COMMODORE USERS

Ahoy!

$2.95/ CAN. $3.35
OCTOBER 1988




The wait is over...
but the excitement has just begun.
More than two years in the making, Warriors of Destiny sets the new standard for role-playing games. No other game can offer you as beautifully detailed a world or as exciting an adventure. Ultima V is the game for all role-playing gamers: Warriors of Destiny will enchant the novice and challenge the expert.
Highlights of Ultima V
• Powerful new combat and magic systems, increasing in sophistication as your skill increases.
• Over twice the graphic detail, animation and sound effects of earlier games.
• Hundreds of truly life-like people who can carry on complex conversations and interactions with you.
Ultima V
Warriors of Destiny
LOUBET '87
Screen displays shown from the Apple™. Others may vary. Allows the use of Ultima IV characters. No previous Ultima experience required.
Reader Service No. 240
ORIGIN SYSTEMS INC.
136 Harvey Road
Building 'B'
Londonderry, NH 03053
Available on Apple and Commodore

Ahoy!

CONTENTS

DEPARTMENTS

FEATURES

PROGRAMS

Cover art by James Regan; photography by Christopher Gould and Michael R. Davila

Commodore owners – save on computer software, hardware, and accessories with the *Ahoy!* Access Club! See page 53 for details.

President
Michael Schneider

Publisher
David Allikas

Executive Editor
Michael R. Davila

Art and Production Director
Laura Palmeri

Technical Editors
David Barron
Bob Lloret

Consulting Editors
Morton Kevelson
Tim Moriarty
Dale Rupert

Entertainment Editor
Arnie Katz

Art Production
Christopher W. Carter

Circulation Director
W. Charles Squires

Advertising Director
Lynne Dominick

Director of Promotion
Mark Kammerer

Promotion Art Director
Stacy Weinberg

Controller
Dan Tunick

Advertising Representative
JE Publishers' Representative
6855 Santa Monica Blvd.
Suite 200
Los Angeles, CA 90038
(213) 467-2266
Boston (617) 437-7628
Dallas (214) 660-2253
New York (212) 724-7767
Chicago (312) 445-2489
Denver (303) 595-4331
San Francisco (415) 864-3252

ISSUE NO. 58 OCTOBER 1988

Ahoy! (ISSN #8750-4383) is published monthly except March, May, July, and September by Ion International Inc., 45 W. 34th St., Suite 500, New York, NY 10001. Subscription rate: 8 issues for $16.95, 24 issues for $31.95 (Canada and elsewhere $22.95 and $42.95 respectively). Second class postage paid at New York, NY 10001 and additional mailing offices.

Postmaster, send address changes to *Ahoy!*, 45 W. 34th Street, Suite 500, New York, NY 10001. Direct all address changes or matters concerning your subscription to *Ahoy!*, P.O. Box #341, Mt. Morris, IL 61054 (phone: 815-734-4151). All editorial inquiries and products for review should be sent to *Ahoy!*, 45 W. 34th St., Suite 500, New York, NY 10001.

Lyco Computer

Marketing & Consultants

Since 1981

Air orders processed within 24 hours.

Lyco Means Total Service.



Mark "Mac" Bowser, Sales Manager

I would personally like to thank all of our past customers for helping to make Lyco Computer one of the largest mail order companies and a leader in the industry. Also, I would like to extend my personal invitation to all computer enthusiasts who have not experienced the services that we provide. Please call our trained sales staff at our toll-free number to inquire about our diverse product line and weekly specials.

First and foremost our philosophy is to keep abreast of the changing market so that we can provide you with not only factory-fresh merchandise but also the newest models offered by the manufacturers at the absolute best possible prices. We offer the widest selection of computer hardware, software and accessories.

Feel free to call Lyco if you want to know more about a particular item. I can't stress enough that our toll-free number is not just for orders. Many companies have a toll-free number for ordering, but if you just want to ask a question about a product, you have to make a toll call. Not at Lyco. Our trained sales staff is knowledgeable about all the products we stock and is happy to answer any questions you may have. We will do our best to make sure that the product you select will fit your application. We also have Saturday hours — one more reason to call us for all your computer needs.

Once you've placed your order with Lyco, we don't forget about you. Our friendly, professional customer service representatives will find answers to your questions about the status of an order, warranties, product availability, or prices.

Lyco Computer stocks a multimillion dollar inventory of factory-fresh merchandise. Chances are we have exactly what you want right in our warehouse. And that means you'll get it fast. In fact, orders are normally shipped **within 24 hours. Free shipping on prepaid orders over $50, and there is no deposit required on C.O.D. orders.** Air freight or UPS Blue/Red Label shipping **is available, too. And all products carry the full manufacturers' warranties.**

I can't see why anyone would shop anywhere else. Selection from our huge in-stock inventory, best price, service that can't be beat—we've got it all here at Lyco Computer.

TO ORDER, CALL TOLL-FREE: 1-800-233-8760
New PA Wats: 1-800-233-8760
Outside Continental US Call: 1-717-494-1030

Hours: 9AM to 8PM, Mon. - Thurs.
9AM to 6PM, Friday — 10AM to 6PM, Saturday

For Customer Service, call 1-717-494-1670,
9AM to 5PM, Mon. - Fri.
Or write: Lyco Computer, Inc.
P.O. Box 5088, Jersey Shore, PA 17740

VISA MasterCard

C.O.D. Risk-Free Policy: • full manufacturers' warranties • no sales tax outside PA • prices show 4% cash discount; add 4% for credit cards • APO, FPO, international: add $5 plus 3% for priority • 4-week clearance on personal checks • we check for credit card theft • sorry, compatibility not guaranteed • return authorization required • due to new product guarantee, return restrictions apply • price/availability subject to change • prepaid orders under $50 in Continental US, add $3.00

1-800-233-8760

Monitors

Thomson:

230 Amber TTL/12" $69.95*
4120 CGA $199.95
4160 CGA $199.95

Blue Chip:

BCM 12" Green TTL ... $64.95
BCM 12" Amber TTL ... $69.95

NEC

Multisync II $589.95

Magnavox:

BM7652 .... $79.95
BM7622 .... $79.95
7BM-613 .... $79.95
7BM-623 .... $79.95
CM8502 .... $179.95
CM8505 .... $199.95
9CM-053 .... $CALL
CM8762 .... $245.95
8CM-515 .... $259.95
CM9043 .... $CALL
8CM-873 .... $499.95

Modems

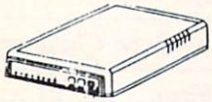
Avatex:

1200e $69.95*
1200i PC Card $65.95
1200hc Modem $89.95
2400 .... $149.95
2400i PC Card $139.95
*w/cable purchase

Hayes:

Smartmodem 300 ...... $139.95
Smartmodem 1200 .... $279.95
Smartmodem 2400 .... $419.95

Avatex™ 2400



$149.95
with cable purchase

COMMODORE

Access:
Echelon $25.95
Mach 5 .... $19.95
Mach – 128 .... $28.95
10th Frame $22.95
Triple Pack $11.95
Wld. Cl. Leader Brd. ... $22.95
Famous Courses #1 ... $11.95
Famous Courses #2 ... $11.95
Leader Board Pack $12.95

Action Soft:
Up Periscope $18.95
Thunderchopper $18.95

Activision:
Champion. Basketball $19.95
Music Studio $19.95
Leather Goddesses $19.95
Top Fuel Eliminator $17.95
Beyond Zork $25.95
GFL Football $19.95
Gee Bee Air Rally $17.95
Last Ninja $19.95
Might & Magic $22.95
Aliens $19.95
Maniac Mansion $19.95
Ghostbusters $9.95

Batteries Included:
Paperclip III $31.95
Outrageous Pages $31.95

Berkeley Softworks:
Geofile C64 .... $29.95
Geocalc C64 .... $29.95
Geopublish C64 .... $39.95
Geos 64 .... $35.95
Geowrite $29.95
Geos 128 .... $39.95
Geowrite 128 .... $39.95
Geocalc 128 .... $39.95
Geofile 128 .... $28.95
Berkeley TriPak $29.95

Broderbund:
Bank St. Writer $29.95
Carmen San Diego $20.95
Graphic Lib. I, II, III $14.95
Print Shop $26.95
Print Shop Compan. $20.95
Cauldron $17.95
Superbike Challenge ... $12.95
Magnetron $15.95

Electronic Arts:
Roadwars $19.95
Hunt for Red October .. $25.95
Patton vs. Rommel $19.95
Skyfox II $19.95
Tobruk–Clash of Armor $19.95

Epyx:
Fastload $22.95

AMIGA

Access:
World Class Lead. Bd. . $CALL

Activision:
Champ. Basketball $25.95
GFL Football $25.95
Gee Bee Air Rally $22.95

Electronic Arts:
Weaver Baseball $33.95
Return to Atlantis .... $CALL
Ferrari Formula One ... $33.95
Hunt for Red October .. $33.95

Epyx:
Apshai Trilogy $11.95
Winter Games $11.95
World Games $22.95
Destroyer $22.95

Firebird:
Guild of Thieves $25.95
Pawn $13.95
Starglider $25.95

Microprose:
Silent Service $22.95

Sublogic:
Flight Simulator II $31.49
Scenery Disk .... $CALL

IBM

Access:
Wld. Cl. Leader Board $27.95
10th Frame $27.95

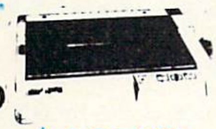
Activision:
Champ. Baseball $22.95
Champ. Basketball $22.95
Zork Trilogy $27.95
Leather Goddesses $22.95
Beyond Zork $27.95

Broderbund:
Ancient Art of War $26.95
Print Shop $34.95
Print Shop Compan. $29.95
Graphic Lib. I or II $19.95
Ancient Art of War at Sea $26.95
Carmen San Diego World $23.95
Superbike Challenge ... $12.95
Search and Destroy $9.95

Electronic Arts:
Yeager's AFT $26.95
Hunt for Red October .. $31.95
Alternate Reality-City .. $25.95

Epyx:
Create A Calendar $15.95
Destroyer $22.95
Winter Games $11.95

STAR micronics

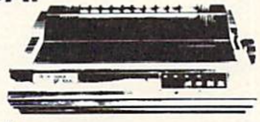


NX-1000
- 144 cps Draft
- 36 cps NLQ
- EZ Font Panel Control

$165.95*
*w/cable purchase

NX-1000 Rainbow Color Printer $225.95

SEIKOSHA SP-180AI



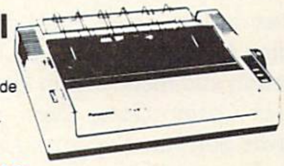
- 100 cps Draft
- 20 cps NLQ
- Std, Par, and IBM Graphics Compatible

Quantities Limited

$125.95

Panasonic Office Automation

1080i Model II



- 150 cps Draft Mode
- NLQ Mode
- Friction & Tractor Feed

$159.95

(2-Year Warranty)

PRINTERS

Seikosha

SP 180Ai	$125.95*
SP 180VC	$125.95*
SP 1000VC	$139.95
SP 1000AP	$159.95
SP 1200VC	$149.95
SP 1200Ai	$159.95
SP 1200AS RS232	$159.95
SL 80Ai	$289.95
MP5420FA	$999.95
SP Series Ribbon	$7.95
SK3000 Ai	$339.95
SK3005 Ai	$419.95
SPB 10	$CALL
SL 130Ai	$599.95

*Quantities Limited

Interfacing available for IBM C 64, Apple and Atari

Panasonic

1080i Model II	$159.95
1091i Model II	$189.95
1092i	$299.95
1592	$375.95
1595	$419.95
3131	$289.95
3151	$459.95
KXP 4450 Laser	$1649.95
1524 24 Pin	$529.95
Fax Partner	$579.95
Optical Scanner	$859.95

Epson

LX800	$184.95
FX1850	$CALL
FX1050	$CALL
EX800	$439.95
LQ500	$339.95
LQ2500	$789.95
GQ3500	$LOW
LQ850	$525.95
LQ1050	$699.95

STAR

NX-1000	$165.95*
NX-1000C	$169.95
NX-1000 Color	$225.95
NX-1000C Color	$229.95
NX-15	$289.95
NR-10	$319.95
NR-15	$419.95
NB-15 24 Pin	$669.95
NX-2400	$309.95
NB24-10 24 Pin	$399.95
NB24-15 24 Pin	$545.95
Laser 8	$1759.95
ND-15	$349.95
NL-10	$149.95

*w/cable purchase

Toshiba

321SL	$489.95
341 SL	$659.95
P351 Model II	$899.95
351 SX 400 cps	$979.95

Citizen

120 D	$145.95
180 D	$159.95
MSP-40	$279.95
MSP-15E	$309.95
MSP-50	$369.95
MSP-45	$399.95
MSP-55	$469.95
Premiere 35	$445.95
Tribute 224	$589.95
Tribute 124	$439.95

Brother

M1109	$159.95
M1509	$335.95
M1709	$459.95
Twinwriter 6 Dot & Daisy	$899.95
M1724L	$619.95
HR20	$345.95
HR40	$559.95
HR60	$649.95

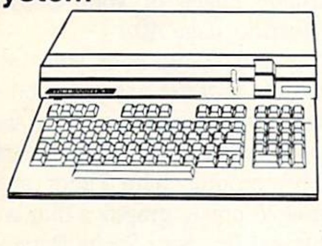
Okidata

Okimate 20	$129.95
Okimate 20 w/cart	$189.95
120	$189.95
180	$219.95
182	$209.95
182 +	$225.95
183	$239.95
192 +	$339.95
193 +	$439.95
292 w/interface	$449.95
293 w/interface	$585.95
294 w/interface	$789.95
393	$955.95
Laser 6	$CALL
390	$479.95
391	$649.95
320	$345.95
321	$445.95

Color Ribbons Available

COMMODORE 128 D System

$439.95



Commodore 128D Plus Thomson 4120 Monitor $649.95

COMMODORE HARDWARE

64C Computer	$149.95
C128D Computer/Drive	$439.95
1541 II Disk Drive	$175.95
1581 Disk Drive	$189.95
Excel 2001 C128 Drive	$199.95
Excel FSD-2+ C64 Drive	$149.95
1802C Monitor	$189.95
1084 Monitor	$299.95
C1351 Mouse	$39.95
1700 RAM	$109.95
1764 RAM C64	$117.95
Colt PC	$689.95

COMMODORE 64C System Special

$459.95

- Commodore 64C Computer
- Excel FSD-2 Disk Drive
- Star NX1000C Commodore Ready Printer



Color System

HEADSTART by Vendex

- Ready to plug in and use immediately
- Ultra fast 8 Mhz Intel 8088-2 Processor
- 512K RAM memory expandable to 768K
- 2-360K disk drives standard
- First complete system with clock calendar and built-in ports for printer, RS232, 2 joysticks, mouse and light pen
- Includes $500 worth FREE software programs
- Hi Res color monitor included!



$989.95



Attention Educational Institutions:

If you are not currently using our educational service program, please call our representatives for details.

COMMODORE COLT PC



- IBM PC Compatible
- 640K Std.
- Two 5¼ Drives Std.
- Expansion for Hard Drive
- Turbo Processor
- MSDOS + GW Basic Included

$689.95

- Serial + Parallel Ports
- Mono/RGB Color Card Included

Circle #268 on Reader Service Card

VIEW FROM THE BRIDGE

If you're not a subscriber, there's a page missing from your magazine. Is it important? You decide:

Are you thinking of placing an order with Rent-A-Disc in this issue? You'd pay 25% less if you had that missing page. Or taking Q-Link up on their offer of a free modem and membership kit? They'd throw in a free *Rabbitjack's Casino Game Disk*–*if* you had that missing page. Maybe Software Support's *Kracker Jax* series caught your eye. You could buy two and get one free if–you guessed it!

The October edition of the *Ahoy! Access Club Clipper* includes these and other special offers from *Ahoy!* advertisers. Some let you save money, some allow you to get more for the same money–but all are special deals available exclusively to Club members!

But don't feel left out. Joining the Club couldn't be easier–see page 53 for details. (Yes, you'll need your checkbook handy–but your only charge will be for the subscription to *Ahoy!* that you meant to take out long ago. Club membership is free!)

Note also that while this month's *Clipper* is a single double-sided page, future editions will be considerably larger, with more mail order offers, lists of dealers around North America offering discounts, and much more. Do you really want to miss all that in the months ahead–and pay more for *Ahoy!* on the newsstand? Naaaah!

Now let's turn to the rest of the October *Ahoy!* (If *that's* missing, please call us right away!)

• The flow of the new non-entertainment software for the 64/128 has slowed to a trickle, but one of the year's more significant new programs appeared in prerelease form at press time. Almost any other product would have been slated for the following month, but this was the eagerly anticipated *GEOS 2.0.* Morton Kevelson gave up his weekend to get his review into this issue. (Turn to page 49.)

• The photo that accompanied last month's *Rupert Report* depicted crystal-clear spheres forming into a cube in midair–an artistic interpretation of Dale Rupert's step-by-step construction of a *Lightning Speed Compiler.* In this month's photo, the cube has come to rest in the plain of spheres below, radiating light in all directions–representing the way compilers (like Dale's *Mini-comp* in this issue) effect a *BASIC Metamorphosis,* changing your programs into high-speed machine language. And you thought we only published artwork because it looked good! (Turn to page 32.)

• Michael A. Grotton had to travel a long way–to Holland, to be exact–to get the inspiration for *Side Border Scroll Editor.* Because he shares his discovery with you in this issue, you don't have to travel as far. But *Side Border Scroll Editor* will take you to places you've never been–the hitherto unusable edges of your C-64 screen. (Turn to page 40.)

• You've told us in your letters that there's nothing you like better than game programming utilities written by your favorite *Ahoy!* game programmers. This month, John Fedor shows you how to create graphics that will make players flip with *Sprite Rotator.* (Turn to page 15.)

• We've said it many times–the better the computer game, the more agonizing it is to wait for your turn while you watch your buddy play. So it's fortunate that Mike Hoyt programmed *Capture* for solitaire or head-to-head competition –it's so good that some of you might get hurt wrestling over joysticks. (Turn to page 58.)

Roam around inside–you'll find a whole lot more of interest. Us–we're off to clip some coupons!

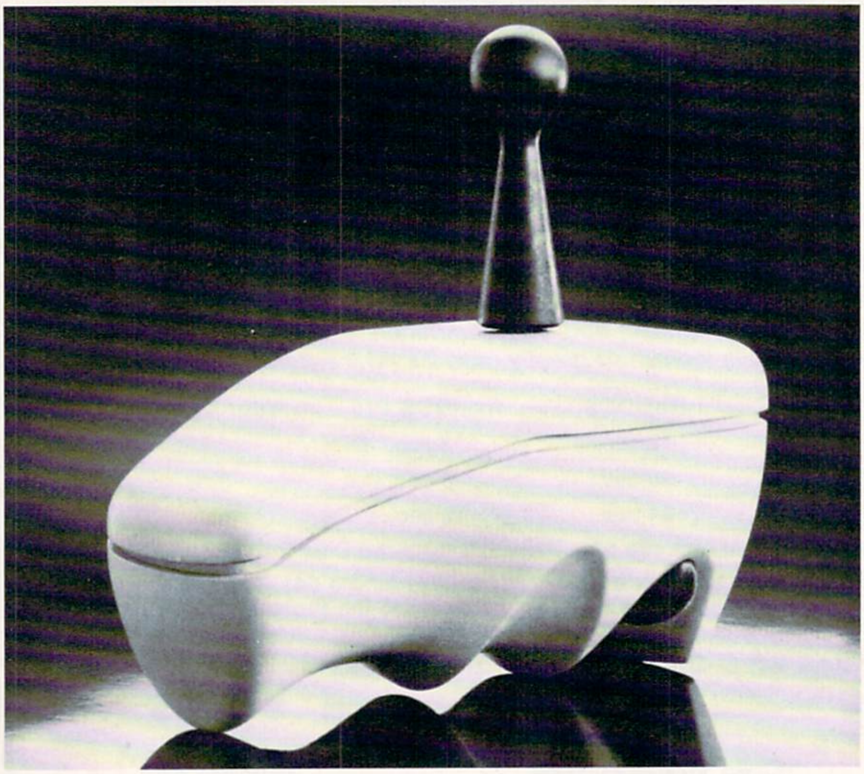
–David Allikas

FREE Q-LINK SOFTWARE AND MORE!

Included free on the back of this month's *Ahoy! Disk* is Q-Link version 4 software, enabling you to log on to the Commodore world's leading online service. Also included free are your first month's membership (regularly $9.95); your first hour of plus time (regularly $4.80); and Q-Link's *Sea Strike* and *Hangman* games, playable online. Use the coupon on page 41 to order the October '88 *Ahoy! Disk.*

SCUTTLEBUTT

SPEECHWRITER • COMMODORE SHOWS • EDUCATIONAL GUIDE • REPAIR MANUALS • MATH PROGRAMS • GAMES FROM DATA EAST, MINDSCAPE, RAINBIRD, FIRST ROW • TYPING TUTOR UPGRADE • ERGONOMIC JOYSTICK • REMOTE COMPUTER CONTROL • READING DISK • FOOTBALL FORECASTER



The callouses that come from gripping a hard plastic joystick for extended periods should be forestalled by Wico's soft, rubberlike Ergostick.

ERGONOMIC JOYSTICK

For use with the 64 among other systems, Wico's Ergostick is form-fit to the human hand with individually sized finger grooves. The soft, pliable material used for the body of the joystick makes it easy to grip. A 90-day limited warranty is included.

Wico Corporation, 312-647-7500 (see address list, page 14)
Circle #275 on Reader Service Card

GEOS WITH 1541

Effective July 1, Commodore will no longer include *GEOS* and *GEOS* applications with the 64. Instead, the *GEOS deskTop* (consisting of *GEOS* kernal, desk accessories, and a variety of fonts) will be included with the 1541 disk drive, and *geoWrite* and *geoPaint*, previously packaged with the 64, will be sold separately.

Commodore Business Machines, 215-431-9100 (see address list, page 14).
Circle #273 on Reader Service Card

FOR YOUR CONVENIENCE...

...every product listed in *Scuttlebutt* will now be accompanied by a Reader Service Number. To save the time and expense of writing directly to individual manufacturers, turn to the Reader Service Card bound between pages 50 and 51 and circle the numbers corresponding to the items you're interested in.

WORLD OF COMMODORE

The Toronto International Center hosts the sixth annual World of Commodore Show December 1-4. Over 20,000 consumers are expected to attend to buy hardware and software, both Commodore and non, from a variety of vendors.

And right in Commodore USA's backyard, at the Philadelphia Civic Center, the first annual World of Commodore Show U.S.A. will take place November 3-6.

The Hunter Group Inc., 416-595-5906 (see address list, page 14).
Circle #274 on Reader Service Card

KEY EXCHANGE

Simon & Schuster will allow owners of their *Typing Tutor III* to upgrade to *IV* by sending a check for $20.00 plus $3.00 postage and handling, along with the original disks, to Microservices (see address list, page 14).
Circle #276 on Reader Service Card

THE ABC'S OF C

A trio of books on C programming: *Advanced C: Tips and Techniques* ($24.95) takes a 456-page look at C, stressing portability, execution efficiency, and programming techniques. Chapters are devoted to debugging techniques, C's run-time environment, and arrays and pointers. Techniques are provided from applications like allocating storage for multidimensional arrays at run-time, working with complex C expressions, and speeding up programs with multidimensional arrays.

Hayden Books, 317-298-5400 (see address list, page 14).
Circle #289 on Reader Service Card

TEVEX Computer Software 1-800-456-1162

SSI

	List $	Our $
B-24	$35	$24
Battlecruiser	$60	$41
Eternal Dagger	$40	$28
Gettysburg	$60	$41
Kampfgruppe	$60	$41
Panzer Strike	$45	$31
Phantasie III	$40	$28
President Elect	$25	$18
Questron II	$40	$28
Realms Darkness	$40	$28
Roadwar 2000	$40	$28
Roadwar Europa	$40	$28
Shard of Spring	$40	$28
Shiloh	$40	$28
Sons of Liberty	$35	$24
Wargame Constr.	$30	$21
War South Pacific	$60	$41
Wizard's Crown	$40	$28

MICROPROSE

	List $	Our $
Airborne Ranger	$35	$24
F-15	$35	$24
Gunship	$35	$24
Pirates	$40	$28
Silent Service	$35	$24
Stealth Fighter	$40	$28

EPYX

	List $	Our $
California Games	$40	$28
Death Sword	$20	$15
Destroyer	$40	$28
4x4 Off Road	$40	$29
Impos. Mis. 2	$40	$29
Movie Monster	$20	$15
Spy vs Spy III	$25	$18
Street Sports Soccer	$40	$28
The Games	$40	$28
World Games	$40	$28

SSG

	List $	Our $
Battles Civil War I	$40	$28
Battles Civil War II	$40	$28
Battles Normandy	$40	$28
Carriers at War	$50	$34
Europe Ablaze	$50	$34
Halls Montezuma	$40	$28
Reach - Stars	$45	$31
Rommel N. Africa	$40	$28
Russia	$40	$28

ACCOLADE

	List $	Our $
Apollo 18	$30	$21
Card Sharks	$30	$21
4th & Inches	$30	$21
Hardball	$30	$21
Plasmatron	$15	$12
Power at Sea	$30	$21
Spy vs Spy I & II	$15	$12
Test Drive	$30	$21
The Train	$30	$21

ACTIVISION

	List $	Our $
Aliens	$35	$24
Last Ninja	$35	$24
Maniac Mansion	$35	$24
Star Rank Boxing	$30	$22

INFOCOM

	List $	Our $
Beyond Zork - 128k	$45	$31
Border Zone	$35	$24
Lurking Horror	$35	$24
Nord & Bert	$35	$24
Sherlock	$35	$24

AMIGA

	List $	Our $
Alien Fires	$40	$28
Alternate Realitry	$40	$28
Breach	$40	$28
Death Sword	$20	$15
Defender - Crown	$50	$34
Empire	$50	$34
FA/18 Interceptor	$50	$34
Ferrari Formula 1	$50	$34
Gettysburg	$60	$41
Hunt-Red October	$50	$34
King of Chicago	$50	$34
Paladin	$40	$28
Ports of Call	$50	$34
Return to Atlantis	$50	$34
Roadwar Europa	$45	$31
Roadwars	$35	$24
S.D.I.	$50	$34
Sinbad	$50	$34
Starfleet I	$55	$38
Three Stooges	$50	$34
Test Drive	$45	$31
Weaver Baseball	$50	$34
World Tour Golf	$40	$28
Zoom	$30	$22

ELEC. ARTS

	List $	Our $
Adv Constr. Set	$15	$12
Bard's Tale I or II	$40	$28
Bard's Tale III	$40	$28
Chess 2000	$40	$28
Chuck Yeager	$35	$24
Dragon's Lair	$25	$18
Hunt Red October	$40	$28
Legacy - Ancients	$30	$21
Lords - Conquest	$15	$12
Marble Madness	$30	$21
Monopoly	$30	$22
Pegasus	$30	$21
Roadwars	$30	$21
Rockford	$30	$22
Skate or Die	$30	$21
Seven Cities	$15	$12
Skyfox II	$30	$21
Strike Fleet	$30	$21
Wasteland	$40	$28

ORIGIN SYS.

	List $	Our $
Auto Duel	$50	$34
Ogre	$30	$21
Ultima I	$40	$28
Ultima III	$50	$34
Ultima IV	$60	$41

ETC.

	List $	Our $
AR - Dungeon	$40	$28
Choplifter-Magic	$15	$12
Concentration	$15	$12
Defender - Crown	$35	$24
Echelon	$45	$31
Full Count Baseball	$40	$30
Gauntlet	$35	$24
High Seas	$50	$34
Jinxster	$35	$25
Might & Magic	$40	$28
ML Baseball	$40	$28
ML Wrestling	$30	$21
NBA	$40	$28
President is Missing	$25	$19
Road Runner	$35	$24
Sinbad	$35	$24
Starfleet I	$40	$28
Stealth Mission	$50	$34
Three Stooges	$35	$24
Under Fire	$35	$24
Up Periscope	$30	$21
Wizardry I	$40	$28
WC Leader Board	$40	$28
Wooden Ships	$35	$24

EDUCATIONAL

	List $	Our $
Carmen - U.S.	$40	$28
Carmen - World	$35	$24
Early Games	$35	$24
Easy as ABC	$40	$28
Fraction Factory	$30	$21
Kindercomp	$21	$15
Magic Spells	$40	$28
Math Blaster	$50	$34
Piece of Cake Math	$35	$24
Reader Rabbit	$40	$28
Rocky's Boots	$35	$24
Speed Reader II	$50	$34
Spell It	$50	$34
SB Spellgrabber	$30	$21
Stickybear ABC	$30	$21
SB Math II	$30	$21
SB Numbers	$30	$21
SB Opposites	$30	$21
SB Reading	$30	$21
Word Attack	$50	$34

PRODUCTIVITY

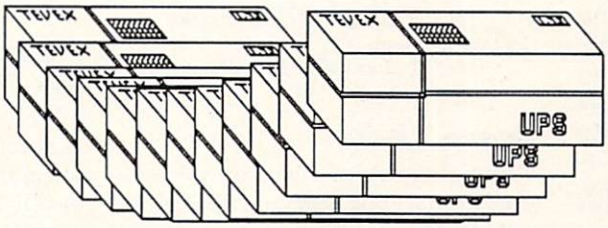
	List $	Our $
Certificate Maker	$40	$28
GEOS - 64	$60	$41
GEOS - 128	$70	$47
Geos Desk Pack I	$35	$24
GeoCalc	$50	$34
GeoDex	$40	$28
GeoFile	$50	$34
GeoProgrammer	$70	$47
GeoPublish	$70	$47
GeoSpell	$30	$21
GeoWrite Wkshop	$50	$34
Newsroom	$50	$34
Paper Clip III	$50	$34
Print Shop	$45	$31
Print Shop Comp.	$35	$24
Toy Shop	$30	$21

ACCESSORIES

Sony DSDD 5.25	$9
Tevex DSDD w/library case	$8
Sony DSDD 3.5	$20
3-M DSDD 3.5	$22
Cleaning Kit 5.25	$6
Cleaning Kit 3.5	$6
EPYX 500 Joystick	$15
Mouse Pad	$6
Safe Strip 6 Surge Protector	$22
Storage Case 60 5.25	$9
Storage Case 40 3.5	$9
Universal Printer Stand	$22
Universal System Stand	$20

Same Day Shipping

Just call before 3:30 and we'll ship your order today by UPS. Your package is only days away with Tevex.



COMING SOON

- Battles of Napoleon
- Covert Action
- Dive Bomber
- The Games - Summer Ed.
- Heroes of the Lance
- Pool of Radiance
- Red Storm Rising
- S.D.I.
- Sporting News Baseball
- Ultima V

CALL TOLL-FREE 1-800-456-1162

FREE CATALOG WITH YOUR FIRST ORDER

Open 9-8 Mon. - Fri. 10-5 Sat.

Retail Store open same hours.

Georgia residents call 404-934-5059. When ordering by mail send money order. Include phone number. SHIPPING: Add $3.00 for shipping and handling charge. Georgia residents add 4% sales tax. Shipping for Canadian orders is 5% of order, with a $4.00 minimum . U. S. Mail, APO & FPO orders add 5% of order, with a $4.00 minimum. Shipping for all other foreign orders is 15% of order, with a $10.00 minimum. All sales are final.

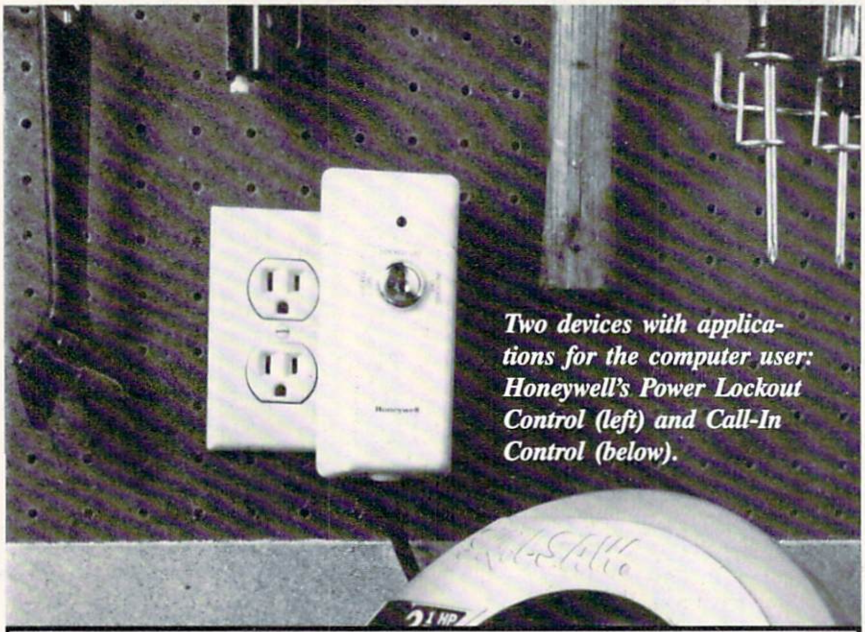
New Titles are underlined

TEVEX

4205 First Ave, Suite 100
Tucker (Atlanta), GA 30084
404-934-5059

VISA MasterCard

Circle #271 on Reader Service Card





Two devices with applications for the computer user: Honeywell's Power Lockout Control (left) and Call-In Control (below).

Artificial Intelligence Programming in C offers an introduction to the subject and a library of over 100 programs and functions (also available on disk) illustrating all aspects of artificial intelligence and enabling programmers to build their own expert system. 260 pages; $17.95 (paperback).

TAB Books Inc. (see address list, page 14).
Circle #290 on Reader Service Card

The C Programming Language, Second Edition is based on the draft-proposed ANSI C Standard now being developed to standardize and modernize the language. New features from the ANSI standard are delineated. Also added are a C reference manual and an appendix describing the standard library. 284 pages; $40 clothbound, $28 paperback.

Prentice Hall, 201-592-2000 (see address list, page 14).
Circle #291 on Reader Service Card

IT'S A LOCK

You can prevent unauthorized persons from getting access to your computer by locking Honeywell's new Power Lockout Control onto its cord. The device can also be used for preventing a child from playing with power tools, monitoring that same rotten kid's TV or VCR viewing, and other uses with any tool or appliance up to 1650 watts. 6KV surge protector and a noise filter are built in.

Honeywell Inc., 612-542-3339 (see address list, page 14).
Circle #280 on Reader Service Card

REMOTE POSSIBILITIES

Also from Honeywell comes the Call-In Control ($79.95), making it possible to turn on or turn off your computer (or air conditioner, crock pot, entry light, etc.) from a remote location. You plug the Control into any outlet, and then plug the appliance you wish to operate and a touch tone phone into the Control. Then call home and let the phone ring 6, 9, or 12 times; when it stops ringing, the phone will beep twice if the appliance is on or once if it's off. By using the buttons on your phone you can start, shut off, or monitor the appliance.

Honeywell Inc., 612-542-3339 (see address list, page 14).
Circle #281 on Reader Service Card

GAMES

All the following games are for the C-64.

As *Karnov* ($29.95), a fire-breathing Russian in search of the Lost Treasure of Babylon, you run, swim, jump, climb, and fly through nine levels of gameplay while battling enemies of various sizes and strengths. You have 10 special combat-enhancing options to choose from, including super-jumping boots, bombs, ladders, boomerangs, wings, shields, and more. But none of them will help you against the Boss Enemy, who can only be destroyed by a direct hit. At the end of each level, you receive a clue to help you complete the next.

Data East USA Inc., 408-286-7074 (see address list, page 14).
Circle #293 on Reader Service Card

Mediagenic (née Activision) has added four previously released titles to its Solid Gold line of $14.95 software for the C-64: *Aliens: The Computer Game,* Infocom's *Leather Goddesses of Phobos,* and Gamestar's *GBA Championship Basketball: Two-on-Two* and *Championship Baseball.*

Mediagenic, 415-329-0500 (see address list, page 14).
Circle #294 on Reader Service Card

Three games to be distributed by Electronic Arts under its affiliated labels program:

Paragon's *Master Ninja: Shadow Warrior of Death* sends you into the 25-

chamber castle of a Japanese warlord in search of a stolen magic sword. Martials arts skills (20 moves are provided) and historic ninja weapons (blow dart, bow and arrow, stars, blinding powder, knives) are needed to overcome samurai guards, deadly tigers, black magic curses, and more. The large size of the characters (over 3½" tall) makes the action easy to follow.

Paragon's *Master Ninja: Shadow Warrior of Death* sends you into the 25-chamber castle of a Japanese warlord in search of a stolen magic sword. Martials arts skills (20 moves are provided) and historic ninja weapons (blow dart, bow and arrow, stars, blinding powder, knives) are needed to overcome samurai guards, deadly tigers, black magic curses, and more. The large size of the characters (over 3½" tall) makes the action easy to follow.

DataSoft's *Cosmic Relief: Prof. Renegade to the Rescue* ($24.95) sends players in quest of K.K. Renegade, who 40 years ago predicted the arrival of a huge asteroid. Now that the space rock has been sighted, the Professor may be the only man who can save the Earth. You must surmount such bizarre obstacles as stone snakes, reptilian birds, and acid storms, as well as find and utilize such curiosities as unicycles and vacuum cleaners to complete your task. Once you find Renegade, you'll help him build an anti-asteroid deflector.

Adapted by Leisure Genius from the board game, *Risk* allows Commodore imperialists to play against friends or against computer opponents who employ a variety of strategies. All the popular game variations are possible, such as US, UK, and short game.

Electronic Arts, 415-571-7171 (see address list, page 14).
Circle #295 on Reader Service Card

Consisting of 20 different and increasingly difficult levels, *Warlock* ($29.95) nonetheless requires minimal hand-eye coordination. Instead, quick payoffs and endearing characters encourage novice to advanced players to continue the action/adventure until the Warlock achieves his goal of finding the stolen Karna.

Three-Sixty, Inc., 408-879-9144 (see address list, page 14).
Circle #296 on Reader Service Card

Mindscape will distribute C-64 adaptations of several SEGA coin-op hits. *Out Run* and *Space Harriet* will be released this fall, with *After Burner, Thunderblades,* and other titles to follow.

Mindscape, 312-480-7667 (see address list, page 14).
Circle #297 on Reader Service Card

For the 64 from Rainbird:

Enlightenment ($19.95) concerns the return of the evil wizard Acamantor to Belorn, 103 years after he was banished by Hasrinaxx the Druid. To defeat him once again, Hasrinaax must make his way through the ten lands of Belorn and Acamantor's five level dungeon, aided by the powers of earth, air, water, and fire.

Black Lamp ($19.95) follows Jolly Jack the Jester's quest to win the princess's hand by defeating a forest full of skull-dropping buzzards, spitting witches, and other beasties, and ultimately taking the black lamp away from the fire breathing dragon that guards it.

Scheduled for fourth quarter release: *The Universal Military Simulator* ($39.95) makes it possible to recreate the battles of Gettysburg, Arbella, Hastings, Marston Moor, and Waterloo, with complete and accurate locales, geographical features, combat troops, and armaments. A battle can be recreated as it occurred, or the original parameters can be altered to explore "what if" scenarios. The program's 3-D graphics system lets you view the action from any angle.

Carrier Command ($34.95) puts you at the bridge of a futuristic vessel as you attempt to capture the enemy's islands and destroy its forces. You're aided by a squadron of remote fighters and an amphibious assault division, up to four of each controllable at once. You protect your ship with defense drones and a 360 degree turret mounted laser cannon with telephoto tracking.

Rainbird, 415-322-0412 (see address list, page 14).
Circle #298 on Reader Service Card

Sir-Tech continues its Wizardry series with *Wizardry II: The Knight of Diamonds* ($39.95), sequel to December's *Wizardty I: Proving Grounds of the Mad Overlord.* The adventurer must find over 100 items and battle more than 100 monsters, in groups of as many as five at a time. Characters developed in *Proving Grounds* are used to maneuver in the 6-level 3-D maze of *Knight.* Average playing time is 50+ hours.

Sir-Tech Software, Inc., 315-393-6633 (see address list, page 14).
Circle #299 on Reader Service Card

Based on the novel of the same name (a copy of which is included), *Not a Penny More, Not a Penny Less* ($29.95) requires the player to solve a series of puzzles that will lead to the recovery of $1,000,000 swindled by a con man. You must bring three fellow victims together and orchestrate a "countersting" that will take you to Monte Carlo, London, Ascot, Wimbledon, and Oxford. A hint sheet will help you get back on track if you get lost.

Simon & Schuster, 212-373-8882 (see address list, page 14).
Circle #300 on Reader Service Card

Terror contains four graphic/text adventures on two disks: *Dracula, Frankenstein, Wolfman,* and *Jack the Ripper.* The documentation features the four stories rewritten to follow the events of the games, to aid frustrated players.



Available to present users of the **Accelerated Reader** *package (shown) for an additional $50, Readup's first in a series of Supplemental Diskettes adds tests for 50 more books including* **A Wrinkle in Time, Call of the Wild, The Black Stallion,** *and more.*

Super Tanks lets players drive three futuristic tanks: *Traxxon, Lifeforce,* and *Stratton.* Each vehicle is designed for a different mission, allowing players to concentrate on infiltrating, destroying, or escaping from any of three scenarios.

Scorpion, 201-663-0202 (see address list, page 14).
Circle #101 on Reader Service Card

An Advanced Battle Disk for use with Lucasfilm/EA's *PHM Pegasus* includes 10 new battle scenarios that pit the NATO hydrofoil against terrorists and Eastern bloc armadas in political hotspots around the globe. Among the new missions are Missile Alley (exchange hostilities with a fleet of Soviet Komar missile ships), Turkey Shoot (face Soviet Hormone Helicopters that outnumber you 12 to 1), Pegasus Vice (intercept drug-running speedboats), Sink the Bismarck (match your speed against the firepower of the famed German battleship), and Jihad II (escort a cargo ship as every maniac between the Port of Kuwait and the Straights of Hormuz tries to deep-six you). Unavailable at retail, the disk can be ordered directly by sending a check or money order for $19.95 (postage included) to Lucasfilm Games, or writing or calling with a VISA or MasterCard number.

Lucasfilm Games, 415-662-1902 (see address list, page 14).
Circle #102 on Reader Service Card

Adaptations of two Konami arcade titles:

Contra pits mortals against intergalactic badman Red Falcon in a guerrilla war in the Amazon jungle.

Rush N' Attack sends you, a guerrilla fighter armed only with a knife, behind enemy lines to rescue dozens of POW's hidden in an isolated camp. If you can't imagine how this is possible, give Sylvester Stallone a call.

Konami, 312-595-1443 (see address list, page 14).
Circle #103 on Reader Service Card

For the 64 from First Row, $29.95 each:

Prime Time lets players see what it's like to run a TV network, buying and cancelling shows, doing lunch, and saying things like "Sweetie-Baby-Cookie."

Moses: The Old Testament #1 lets the player follow Moses' mission in a lighthearted but historically accurate graphic adventure designed for the novice.

First Row Software Publishing, Inc., 215-337-1500 (see address list, page 14).
Circle #104 on Reader Service Card

READING TESTS

Readup has made available the first of a series of Supplemental Diskettes for use with *The Accelerated Reader,* which tests students' reading comprehension and accumulates results for each child and for the class as a whole. The $50 diskette adds tests for 50 books to the original program's list of 200. The original package is priced at $300 for the C-64.

Readup Inc., 715-887-2333 (see address list, page 14).
Circle #286 on Reader Service Card

$14.95 SOFTWARE

Two additions to IntraCorp's line of $14.95 software for the 64:

Space Math uses an arcade game to improve math skills. Players 8 and older can compete on 8 levels at 27 difficulty settings.

Pro Football Facts and Predictions, 1988 Edition contains five years of gridiron stats, from 1983 through the January 1988 Super Bowl, with an analysis program to help predict this year's winners.

IntraCorp, 305-252-9040 (see address list, page 14).
Circle #287 on Reader Service Card

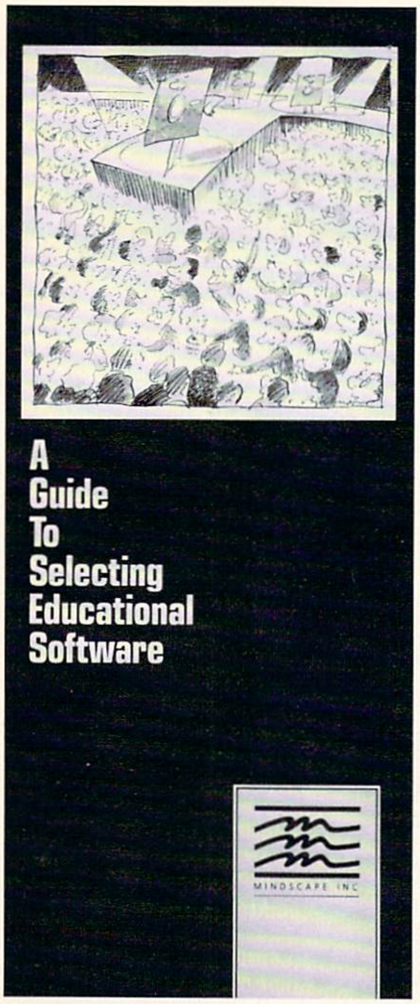
REAL WORLD MATH

Gamco's *Math for Successful Living* series provides students with mathematical situations they are likely to encounter in real life. In *Timecards and Paychecks* ($44.95) they compute overtime, commission, piecework pay, tips, gross and net pay, and various deductions. *Managing a Checking Account* ($44.95) covers making deposits, keeping a check register, and reconciling a monthly statement. *Shopping Strategies* ($74.95) is two disks' worth of planning a menu and buying the groceries required, figuring out unit prices, applying coupons and specials, making change, and determining sales tax.

Gamco Industries, Inc., 800-351-1404; in TX call collect 915-267-6327 (see address list, page 14).
Circle #288 on Reader Service Card

MAKE A SPEECH

Don't let what happened to Joe Biden happen to you. Based on Robert Shelley's *Pocket Speechwriter* book, *Computer Speechwriter* ($39.95) allows you to create a quasi-original speech in minutes on the C-64. If you have a subject in mind, you call up the key word index, find appropriate material,



Free guide to the basics of evaluating and selecting educational software.

and tailor it to your speech; if you have no subject in mind, you can choose from dozens of prewritten speeches on different topics, or combine an opening line, a theme, a few quips, an observation, and a clincher. Included is a speaker's guide full of tips on delivery, timing, preparation, and presentation, covering topics like sizing up an audience and overcoming stage fright. An unconditional money-back guarantee is offered.

Pageant Publishing Co., 514-935-8273 (see address list, page 14).
Circle #277 on Reader Service Card

GUIDANCE DEPT.

In "A Guide to Selecting Educational Software," Mindscape outlines the basics of software evaluation and selection. A free copy of the leaflet is available for a self-addressed, stamped envelope; organizations and institutions may order up to 99 leaflets for 15¢ each, 100-249 for 12¢ each, and 250+ for 10¢ each.

Mindscape, 312-480-7667 (see address list, page 14).
Circle #278 on Reader Service Card

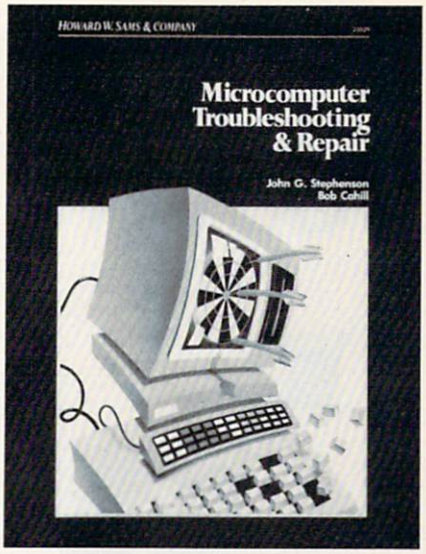
PHOTO CONTEST

In conjunction with Computer Learning Month (which happens to be October), Mindscape is offering prizes of educational software for photographs that capture the experience of kids learning with technology. Awarded will be a grand prize of $500 worth of programs, two second prizes of $250 worth, three third prizes of $100 worth, and ten prizes of one program each. Photos, which become the property of Mindscape, must be submitted by October 31 to Mindscape, c/o Educational Division Photo Contest.

Mindscape, 312-480-7667 (see address list, page 14).
Circle #279 on Reader Service Card

QUICK FIXES

Two computer repair books: *Microcomputer Troubleshooting &*



Troubleshooting strategies for computers, printers, monitors, disk drives.

Repair ($21.95) describes ways of making computer equipment more reliable, and saving time and money on repairs. Basic troubleshooting principles are discussed, along with methods of applying them. Along with computers,

the book covers printers, modems, monitors, disk drives, and interfaces.

Howard W. Sams & Company (see address list below).

Circle #282 on Reader Service Card

Commodore Care Manual: Diagnosing and Maintaining Your 64 or 128 System ($16.95) includes diagnostic program modules that let the user locate problems and determine if the computer is working properly after repairs. For those who don't know BASIC, the programs are explained line by line. Using the book as a guide, Commodore owners can perform preventive maintenance, troubleshoot malfunctions, make simple repairs, and write diagnostic programs for troubleshooting other peripherals. Complete listings for the BASIC programs are given in the appendices, or readers can order the software on disk.

TAB Books Inc., 717-794-2191 (see address list below).

Circle #283 on Reader Service Card

KODACOLOR

10-packs of colored Kodak disks—two each in red, yellow, orange, blue and green—are now available from Verbatim, Kodak's subsidiary. The soft-sectored disks are double-sided, double-density. Consumers who send in a coupon from specially marked packages along with two proofs of purchase will receive a free Kodak Data Highlighter. Suggested list price is $15.50 per 10-pack.

Verbatim News Services, 716-724-5130 (see address list below).

Circle #284 on Reader Service Card

Companies Mentioned in *Scuttlebutt*

Commodore Business Machines
1200 Wilson Drive
West Chester, PA 19380
Phone: 215-431-9100

Data East USA, Inc.
470 Needles Drive
San Jose, CA 95112
Phone: 408-286-7074

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
Phone: 415-571-7171

First Row Software
900 E. 8th Ave.—Ste. 300
King of Prussia, PA 19406
Phone: 215-337-1500

Gamco Industries, Inc.
Box 1911
Big Spring, TX 79721
Phone: 800-351-1404; in TX 915-267-6327

Hayden Books
(c/o **Macmillan, Inc.**)

Honeywell, Inc.
1985 Douglas Drive North
Golden Valley, MN 55422
Phone: 612-542-7500

Howard W. Sams & Co.
(c/o **Macmillan, Inc.**)

IntraCorp, Inc.
14160 S.W. 139th Court
Miami, FL 33186
Phone: 800-INTRACO or 305-252-9040

Konami Inc.
815 Mittel Drive
Wood Dale, IL 60191
Phone: 312-595-1443

Lucasfilm Games
P.O. Box 2009
San Rafael, CA 94912
Phone: 415-662-1966

Macmillan, Inc.
4300 W. 62nd Street
Indianapolis, IN 46268
Phone: 317-298-5400

Mediagenic, Inc.
3885 Bohannon Drive
Menlo Park, CA 94025
Phone: 415-329-0500

Microservices
200 Old Tappan Road
Tappan, NJ 07675
Phone: 800-624-0023; in NJ 800-624-0024

Mindscape, Inc.
3444 Dundee Rd.
Northbrook, IL 60062
Phone: 312-480-7667

Pageant Publishing Co.
P.O. Box 1288
Champlain, NY 12919
Phone: 514-935-8273

Prentice Hall
Prentice Hall Building
Englewood Cliffs, NJ 07632
Phone: 201-592-2000

Rainbird Software
3885 Bohannon Drive
Menlo Park, CA 94025
Phone: 415-322-0412

Readup Inc.
P.O. Box 95
Port Edwards, WI 54469
Phone: 715-887-2333

Scorpion
19 Harbor Drive
Lake Hopatcong, NJ 07849
Phone: 201-663-0202

Simon & Schuster Software
Gulf+Western Building
One Gulf+Western Plaza
New York, NY 10023
Phone: 212-373-8882

Sir-Tech Software, Inc.
P.O. Box 245
Charlestown Mall
Ogdensburg, NY 13669
Phone: 315-393-6633

TAB Books Inc.
Blue Ridge Summit, PA 17294-0850
Phone: 717-794-2191

The Hunter Group, Inc.
204 Richmond Street West, Suite 410
Toronto, Ontario
M5V 1V6 Canada
Phone: 416-595-5906

Three-Sixty, Inc.
2105 South Bascom Ave.
Campbell, CA 95008
Phone: 408-879-9144

Verbatim News Services
Eastman Kodak Company
343 State Street
Rochester, NY 14650
Phone: 716-724-5130

Wico Corporation
6400 W. Gross Point Rd.
Niles, IL 60648
Phone: 312-647-7500

SPRITE ROTATOR

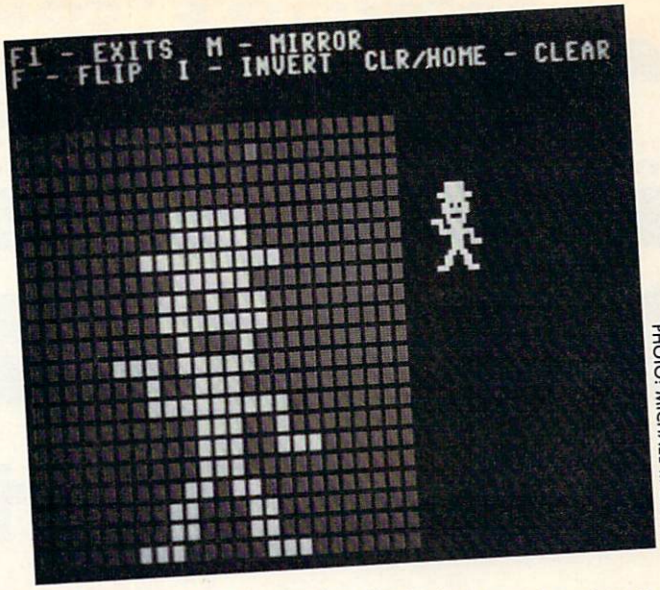


PHOTO: MICHAEL R. DAVILA

For the C-64

By John Fedor

In programming your own special game, you may need to create an object that appears to rotate in shape. For example, a stick thrown to a dog must look like it is turning end over end. While doing this manually is cumbersome and the results may not be accurate, doing the same project with this program will be easier and much more accurate.

Sprite Rotator lets you take a shape (from another sprite creator, or the C-128's built-in one) and rotate it any number of degrees. The program contains eight routines built in for use. Each function is centered around two sprites: a source and a destination. In order for certain routines to function properly, the source or destination sprite must be preset. For others, the source or destination sprite might be changed in some way or another. The particular situation is described for each below.

To load a file of shapes, choose option 1. Type in the name of the file and the computer will load in the shapes. The source sprite will now point to the beginning of the shapes just loaded.

To change which shape the source sprite points at, choose the second option. By using the onscreen instructions, you can easily change the source sprite's pointer. If you need to change the destination pointer, choose the third option which follows the same guideline.

To actually rotate a sprite, the source pointer must be preset to point to the original shape which will be rotated. The destination pointer must be preset to point to the place where you would like to put the rotated shape. The current destination shape will be destroyed by calling this routine. Press 4 (for this option) and type in the number of degrees you want the shape to be rotated. Then wait as it rotates the shape. Be certain that the destination sprite is *not* an important shape, because it will be destroyed by the new rotated shape.

Once you have rotated several shapes, you may need to see if the shape change is smooth enough for your purpose. Choose the fifth option to find out. You choose the starting and ending of the animation cycle. The shapes will then flip continuously until you abort the routine. Pressing the + or − key will adjust the speed of the animation.

After the shapes have been created to your liking, the save function will put the shapes onto disk. The source pointer is the first shape put on. All the shapes between the source and destination (inclusive) are also placed onto the disk in the filename that you chose.

If, after you rotated the shape, you are not satisfied with the appearance of the shape, you can call this routine which will allow you to change the shape of the source sprite. Using a joystick in Port 2, move the yellow cursor around the grid. Pressing the button will flip the pixel on or off. Pressing I will invert all the pixels. Pressing M will mirror the shape, while F will flip it. Pressing CLR/HOME will clear the shape. If you do not have a sprite creator, you could use this to create your own shapes. Pressing F1 will exit from this routine.

The last routine that can be called is copy. With the source and destination preset, the source is copied to the destination as soon as you press the key. Again, make sure your destination is where you think it is. Take the extra key press to see if the destination is at the proper address. Better safe than sorry.

A word of warning: beware of the edges. Because your sprite is a 24x21 pixel shape, there is a 3 pixel difference between the X and Y axes. The program disregards any points that fall out of the 24x21 range. These points may have been important to the shape. To fix this problem, make the longer part the Y axis; that way nothing will be lost when rotated.

This utility is made up of two programs. The first is written in BASIC. It contains the main routines. The second is in ML. Anything that was too slow was put into ML, if possible. The sprite rotation routine (which uses trig functions) could not be easily changed to ML, so it was left in BASIC. This program does not handle multicolor shapes properly. But you could make the shape in one color, rotate it, and see what the new shape should look like.

To type in *Sprite Rotator,* type in the BASIC portion and save it to disk. The ML routines need to be entered with *Flankspeed* (page 61). Save as "ROT.ML". When ready to use, load the BASIC portion. The ML routines will be loaded automatically.

This program is quite easy to use, and powerful. Many great games can be created with rotating shapes for special effects. Introduction screens can have that extra "something" to make them more memorable. This utility adds a new dimension to your programs. ☐

SEE PROGRAM LISTING ON PAGE 65

PROTECTO'S "COMPUTER DIRECT"

Is 1st In PRICE, SUPPORT, & GUARANTEE!

WE LOVE OUR CUSTOMERS! Prices Expire 10-31-88

15 Day Home Trial • 90 Day Immediate Replacement Policy

Complete C64c Commodore System



- Commodore 64c Computer
- 1541 II Disk Drive
- 12" Monochrome Monitor
- Genuine IBM® Printer With Interface And One Roll Of Paper
- GEOS Program For Word Processing & Drawing

Our Low Sale Price

$395.95

List $825

C128D Computer

- Full 128 Detached Keyboard
- 1571 Drive With 128K Of User Memory Expandable To 640K
- 3 Operation Modes (C64, C128, & CP/M, Ver. 3.0)
- 40/80 Column Output, 16 colors
- **Free Programmer's Reference Guide**

Our Low Sale Price

$419.95

List $549

C1541 II Disk Drive

- Increase in Speed
- Increase in Storage Capacity
- Increase in Reliability

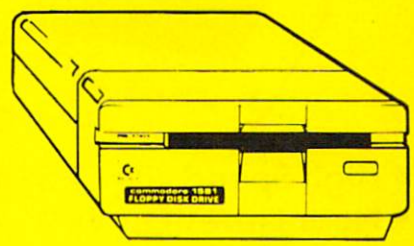
Our Low Sale Price

$159.95

List $299

800K 3½" Disk Drive

Commodore 1581 Works With C64/C64c, and C128/C128D. Over 800K Bytes of Formatted Storage On Double-Sided 3.5" Microdiskettes.



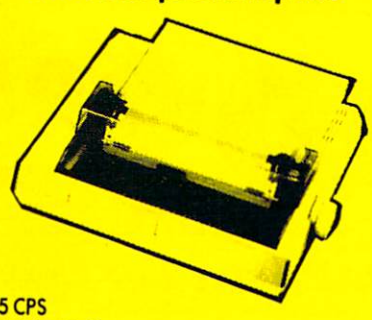
Our Low Sale Price

$179.95

List $249

24-Pin Printer

Document Quality IBM And Epson Graphics



- 135 CPS
- 32K Print Buffer
- Serial Impact Dot Matrix
- 24 Wires (12 x 2 staggered, diameter 0.2 mm)
- IBM and Epson Mode
- Adjustable Sprocket & Friction Feed
- Parallel Centronics Port

Our Low Sale Price

$359.95

List $599

300/1200 BPS External Modem

Capetronic

- **Fully Hayes Compatible**
- 1200 Baud Bell 212A at 1200 or 300 Bits/Sec
- Automatic Dialing & Answering
- Auto Redial At Receipt Of Busy Signal
- Built-In Speaker With Volume Control
- Built-In Help Commands
- Both Touch Tone and Pulse Dialing
- Uses RS-232C Interface (Cable Required)

Our Low Sale Price $79.95

List $139

300/1200/2400 BPS External Modem

Supra Corporation

- **Fully Hayes Compatible**
- 2400 Baud Bell 212A At 300, 1200 & 2400 BPS
- Both Tone & Pulse Dialing
- Built-In Speaker With Volume Control
- Command Buffer That Holds Up To 40 Characters
- Uses RS-232C Interface (Cable Required)

Our Low Sale Price $149.95

List $249

300/1200 BPS Migent Pocket Modem



- Hayes compatible
- Runs on A/C or battery
- Turns on/off automatically
- Small enough to put in your pocket
- Plugs directly into your serial port on many computers
- Easy to use; no switches to set
- On-screen status lights show call progress

Our Low Sale Price

$99.95

List $413

COMPUTER DIRECT

22292 N. Pepper Rd, Barrington, IL. 60010

Call (312) 382-5050 To Order!

* Illinois residents add 6½% sales tax. All orders must be in U.S. Dollars. We ship to all points in the U.S., CANADA, PUERTO RICO, & APO-FPO. Please call for charges outside continental U.S. or C.O.D. **MAIL ORDERS** enclose cashier check, money order, or personal check. Allow 14 days delivery, 2 to 7 days for phone orders and 1 day express mail. Prices, specifications & availability subject to change without notice. Shipping - $3.00 minimum, call for exact charges. Shipping and handling charges are not refundable. (Monitors only shipped in continental U.S.) **COMPUTER DIRECT** will match any valid nationally advertised delivered price on the exact same product with a comparable payment method (excluding any applicable sales taxes).

VISA — MASTERCARD — C.O.D.

Circle #270 on Reader Service Card

PROTECTO'S "COMPUTER DIRECT"

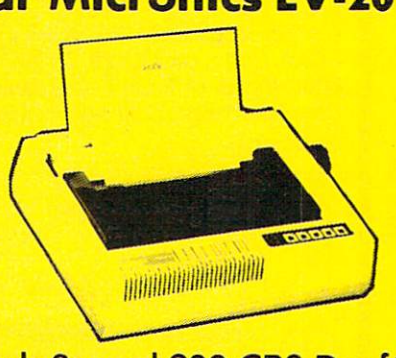
Will Not Be UNDERSOLD!

AND WE MEAN IT!* | Prices Expire 10-31-88

Free Catalog • Over 2,000 Software & Accessory Items In Stock

Super Hi-Speed 200 CPS Printer

Star Micronics LV-2010 10" Printer With Near Letter Quality



Sale $189.95

List $499

No One Sells This Printer For Less!

- High Speed 200 CPS Draft
- Near Letter Quality
- Tractor and Friction Feed
- Automatic Sheet Feed Loading
- Both Parallel & Serial Interface Ports
- Continuous Underline
- IBM Compatible
- Ultra High-Res Bit Image Graphics

Printing Speed
200 Characters Per Second Draft
45 Characters Per Second NLQ

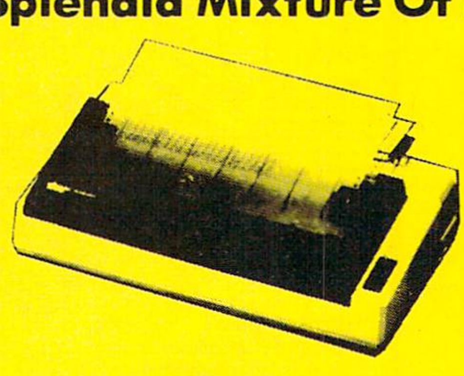
Print Method
Serial Impact Dot Matrix

Print Buffer
16K Bytes

Port
Both Centronics Parallel & Serial Ports

15" High Speed 160 CPS Printer

Splendid Mixture Of High Speed And Near Letter Quality



Sale $209.95

List $499

No One Sells This Printer For Less!

- NLQ and High Draft Speed
- Continuous Underline
- Skip Over Perforation
- IBM Compatible
- Friction and Tractor Feeds
- Both Parallel & Serial Interface Ports
- High Res Graphics Printing

Printing Speed
160 Characters Per Second Draft

Print Method
Serial Impact Dot Matrix

Paper Feed
Sprocket or Friction Feed

Port
Both Centronics Parallel & RS-232 Serial Ports

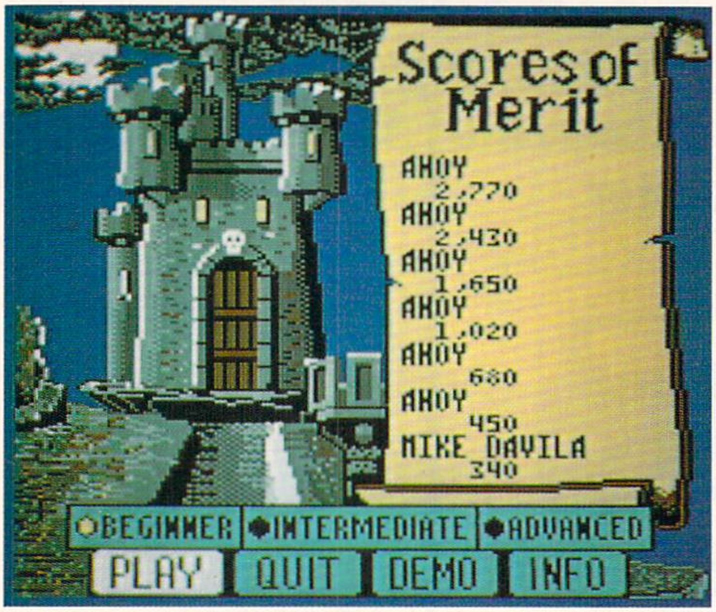
Call **(312) 382-5050**

We Love Our Customers!

Mail

PROTECTO'S COMPUTER DIRECT
22292 N. Pepper Rd., Barrington, IL. 60010

Circle #270 on Reader Service Card



Enter the Dark Castle *and struggle to survive against a variety of cleverly animated creatures and humans by climbing, jumping, and fighting. READER SERVICE NO. 107*

DARK CASTLE
Three-Sixty Pacific
Commodore 64
Disk; $34.95

This fantasy-oriented action game transports the Commodore 64 to two strange worlds which may seem hauntingly familiar. The game concerns the exploration of a creepy mansion inhabited by the Black Knight and his nasty crew. That should certainly be a familiar theme to veteran gamers, though the treatment of the situation is surprisingly fresh and distinctive.

And the other strange world? The realm of Macintosh entertainment software. *Dark Castle*, published by SILICON Beach Software, drew raves from critics and Macintosh owners alike last year, and Three-Sixty evidently hopes to win similar approval from owners of other home computer systems.

They are probably doomed to disappointment. What was fresh on the Mac turns out to be pretty standard stuff for the Commodore 64.

Dark Castle is quite well programmed (by Sculptured Software) and has clever animations for objects like scuttling animals and whip-wielding henchmen. Yet there's little to separate it from a dozen other side-perspective climbing, jumping, and fighting games.

The player controls a hero who enters a haunted castle and battles for survival against a variety of evil creatures and humans. The game begins with the "Great Hall" screen, which displays four possible pathways. Using the joystick, the player positions a cursor over the desired door and presses the action button to send the hero to the play screen which lies behind the selected entrance. Each non-scrolling field shows a side view of several floors of the castle, connected by ladders, staircases, and ropes.

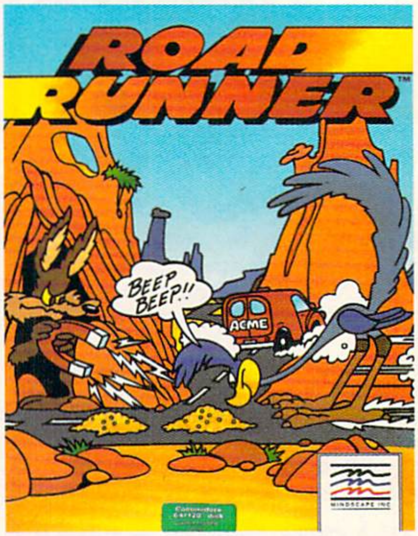
The computerist employs combinations of joystick movement and the action button to make the hero walk in either direction, take three different trajectory jumps, or throw rocks at the numerous antagonists. There is an alternate keyboard-based control system for players who disdain joysticks.

"A rose is a rose is a rose," the poet said, but there is evidently a difference between rules for flowers and computer games. *Dark Castle* does not thrill as deeply in the Commodore universe as it did in Macintosh-land.

Three-Sixty Pacific, 2105 So. Bascom Ave., Campbell, CA 95008 (phone: 408-879-9144). *–Arnie Katz*



Combinations of joystick movement and the action button are used to make the hero walk, jump, and throw rocks at foes. There is an alternative keyboard-based control system as well.



Road Runner: *starve a coyote today. READER SERVICE NO. 108*

ROAD RUNNER
Mindscape
Commodore 64
Disk; $34.95

BeepBeep! Road Runner, speedy

PURE-STAT BASEBALL™



From **Software Simulations** comes an incredibly sophisticated, realistic, and complete statistical baseball simulation for zero, one or two players. Game play, manager's functions, graphics, and a complete statistical library establish Pure-Stat Baseball™ as the ultimate sports simulation program.

Suggested retail price

$39.95

Optional Team and Create Your Own Team Disks Available.

Available for C-64/128, Apple, IBM-PC

Game Play

Pure-Stat Baseball's underlying statistical framework simulates the realities of baseball like never before. Within this framework the program considers each player's batting statistics against both **LEFT**- and **RIGHT**-handed pitchers, where available, and pitchers stats vs. both **LEFT**- and **RIGHT**-handed batters. Every player's fielding and base-running abilities are also considered (an important factor when attempting to steal a base, etc.)

Select the team you'd like to manage, then pick the team you want to play against. Every team from the 1985 season is accurately represented, along with eight classic teams from the past. Determine your starting lineup, designate a starting pitcher, and make player substitutions when necessary. You call the plays, offense and defense. In a one-player game, your computer opponent displays un-canny intelligence in reacting to your managing decisions.

Manager's Functions

Built-in manager's functions provide extra versatility. The Trading function gives you the opportunity to create the "what if" team(s) you've always dreamed about. With this function you can have Dwight Gooden playing with Mickey Mantel and Babe Ruth. Or, if you like, you can form your own draft leagues from existing teams and play against your friends for your own pennant and World Series.

The Stat-Keeper function compiles all of the players' statistics for you and calculates Batting Averages and pitchers' ERAs. You can track your own teams' performances and print the season statistics for your club or league.

Stadium Disk

An optional Stadium Disk is also available that lets you play in any of the twenty-six Major League stadiums.

Statistics

Pure-Stat Baseball™ can maintain a complete statistical record of each team player's performance. You can print all of your players' year-to-date performance statistics at anytime throughout the season. All player stats and game Box Scores can be displayed on the screen or sent to an external printer for a hardcopy printout. The PRINT TEAM STATS function can be used at the end of the season to determine your Most Valuable Player, or to review your club's performance.

Pure-Stat Baseball™ also includes a unique Auto-Play option that lets the computer play a complete game in less than three minutes. A whole series of games can be played unattended, and an entire season of player and team statistics can be compiled over several days with all game stats printed out for your records.

Discover for yourself just how enjoyable a baseball simulation game can be. Pure-Stat Baseball,™ the first real baseball game on a computer.

See your local dealer, or call:

Software Simulations

959 Main Street, Stratford, CT 06497

203/377-4339

and speechless star of numerous classic Warner Brothers cartoons, just arrived on the computer game scene in his perennial cloud of dust, only a little ahead of his frustrated predator Wile E. Coyote. These two adversaries romp through miles of desert terrain, in an abnormally cute multigame arcade-style program.

Developed by programmer Sean Townsend, graphics artist Martin Calvert, and audio man Fred Gray (collectively known as Canvas), *Road Runner* is a visual delight. It captures the rich colors and textures of animator Chuck Jones' Death Valley landscapes as accurately as it reproduces the two sparring stars. Every stretch of roadway is a potential deathtrap, the implications of which have already been grasped by the fertile imagination of "genius" Wile E. Coyote.

The player controls Road Runner via joystick as he zips over the neatly scrolling highway. The game uses a "false" perspective very popular in games of this type: characters are seen from the side while the terrain is displayed from an angled aerial view.

Constantly pursued by the fiendishly clever Coyote, the player must navigate RR over twisting pathways to collect the birdseed scattered along the road (a classic Wile E. ruse!).

The control system consists of compass commands and a jump button, and the games themselves are equally elementary. There are simple mazes, traps, and avoidance games, as well as the basic pursuit contest in which the Road Runner can attempt to lure Wile E. in front of passing vehicles while collecting birdseed (and, later, lemonade).

A window at the top of the screen displays a "Seed Meter," a cute but somewhat awkward device for monitoring the bird's strength. Five bowls of birdseed are arrayed side by side and Road Runner constantly munches on them, starting at the right. As long as the player keeps collecting birdseed, the Road Runner remains at the far right end of the window, indicating optimal strength.

Players start with five lives, and one is lost every time Wile E. catches up with the feathered surrogate. But fear not: the game's star is not slaughtered and devoured onscreen—Wile E. hustles his prize offscreen in order to feast on him discreetly.

For all its charm, however, *Road Runner* is not without flaws. The characters are quite small, and though they are marvelous reproductions, one must peer intently into the screen in order to adequately appreciate that fact. The landscape dwarfs them, which may have been desirable from an aesthetic point of view, but not from a game design perspective.

It's also a shame that there isn't a two-player version of the game where one player could control Road Runner while the other manipulated Wile E. The Coyote acquires all sorts of neat gizmos through the course of the game (a rocket pack, super pogo stick, etc.). These tend to go slightly awry, but work well enough that he remains a threat. It would have been great fun to play this game head to head.

But the major problem here is the creators' failure to adequately communicate the game's dominant element: speed. Although the characters' legs churn furiously, as in the cartoons, the actual onscreen movement is tedious. To compensate, the road is heavy on sharp twists and numerous turns, but the lack of a satisfying velocity keeps the game from generating any real heat.

Road Runner is *very* cute. It contains many of the elements that make the Road Runner-Wile E. Coyote contest a cult favorite. But, larger graphics and the addition of a head-to-head mode would have made this simple contest more than just a pretty face!

Mindscape, 3444 Dundee Rd., Northbrook, IL 60062 (phone: 312-480-7667). *—Bill Kunkel*

4 X 4 OFF-ROAD RACING
Epyx
Commodore 64
Disk; $39.95

Many have described America as a nation on wheels. We are a people who enjoy motoring along the highways and country roads almost as much as reaching our actual destinations. "Getting there is half the fun" is a concept deeply rooted in the American psyche.

One expression of this love affair with vehicles is the tremendous number of driving games and simulations available for home computers. *4 x 4 Off-Road Racing* is the second title involving all-terrain vehicles released for the Commodore this year.

This Ogdon Micro Design creation has much more depth than *Speed Buggy* (Data East), which is essentially a boot and play arcade game. Although the "beginner" skill level facilitates a nearly instantaneous startup, menu-activated modules permit the player to pick a course, select a car, and customize it with handy items like an extra gas tank, a winch, or a camper body.

4 x 4 Off-Road Racing features four obstacle-strewn courses: Baja, Death Valley, Georgia, and Michigan. Each presents the driver with a different set of problems. Rugged terrain and extreme heat are the main challenges on the Death Valley run, while snow, ice, and mud bedevil those who test the Michigan course.

The program evaluates the performance characteristics of each of the four vehicles included in *4 x 4 Off-Road Racing* in seven areas. The Stormtrooper, Tarantula, Highlander, and Katana are rated for power, weight, gas mileage, endurance, ease of repair, payload with cap (camper body), and fuel capacity.

These seven factors exert a strong effect on the race. A heavier truck like the Stormtrooper runs through the gas faster, but it can take more bumps and bangs from rocks, loose tires, competing off-road racers, and other hazards. Matching the car to the course is a key part of the game's strategy.

After the player presses the joystick button to pick a car, the scene shifts to auto parts row, moving the onscreen character to the door of one of the two stores. The Auto Mart vends basic supplies like oil, water, coolant, transmission fluid, batteries, spare tires, repair tools, and a map. Hiring a skilled mechanic to ride shotgun fills up precious cargo space, but it can drastically trim repair times. A six-pack is a frivolous (but potentially satisfying) extra which is also available at this shop.

The Custom Shop carries three grades of tires, winches, extra-capacity fuel tanks, and caps.

The wise driver customizes his or her vehicle to meet the rigors of the chosen route. For instance, mudder

Continued on page 46

TIPS AHOY!

Compiled by Michael R. Davila

Send your best programming and hardware hints to *Tips Ahoy!*, c/o Ion International Inc., 45 W. 34th St.—Suite 500, NY, NY 10001. Include a stamped and self-addressed envelope.

DOWNSHIFT 64

Whoa, pardner! Hit the skids! Downshift, already! Ever feel like that when running or listing a program? Well, now you can vary the speed at which your computer runs with *Downshift 64.* Here's how it works.

With *Downshift 64* installed as your new transmission, the F1 key "shifts" the computer to a lower gear for a slightly slower speed. F3 slows it down even more, and F5 reduces it to a crawl. F7 is like neutral. The program (or list) just sits there while the world continues to rotate on its axis. *Downshift 64* is great for perusing lists, or better yet, tracking program execution to locate bugs. What the heck?! It's fun just to watch Ol' Reliable in slow motion!

First load and run *Downshift 64.* To activate it, type SYS 53000. Now press RETURN. It remains undisturbed in a free area of memory while you load, save, or work on your BASIC programs.

RUN STOP/RESTORE deactivates *Downshift 64.* SYS to the appropriate address to reactivate it.

The program can easily be relocated if you want. Just change the variable A in line 2 to your new starting address. Don't forget to SYS there if you do.

Zip through the gears with *Downshift 64.* And fear not ...you don't need a clutch!

—Buck Childress
Salem, OR

```
·1 REM *** DOWNSHIFT 64 ***
·2 A=53000:PRINTCHR$(147)
·3 FORJ=ATOA+67:READB:POKEJ,B:X=X+B:NEXTJ
·4 IFX<>9903THENPRINT"ERROR IN DATA[3"."]
 ":END
·5 C=INT(A/256):D=(A-(C*256))+13:POKEA+1,
 D:POKEA+3,C
·6 PRINT"DATA OK * SYS"A"TO ACTIVATE[3"."
 ]":END
·7 DATA169,21,162,207,120,141,20,3,142,21
 ,3,88
·8 DATA96,165,203,160,0,201,4,208,8,232,2
 08,253
·9 DATA200,192,64,144,248,201,5,208,8,232
 ,208,253
·10 DATA200,192,128,144,248,201,6,208,8,2
 32,208,253
·11 DATA200,192,192,144,248,201,3,208,8,3
 2,159,255
·12 DATA165,203,184,80,244,76,49,234
```

SCREEN HACK

Below is a program listing that produces a very original and very impressive screen display. I'm not going to tell you exactly what it does; you'll have to type it in to find out. Be sure to save a copy first (a typo in this program can crash your computer megafast). After you've done that, grab hold of something solid and run the program.

```
·10 PRINT"[CLEAR][WHITE]":FORA=49152TO491
 95:READB:POKEA,B:NEXT:POKE2,.:POKE792,.:
 POKE793,192
·20 POKE56580,100:POKE56581,1:POKE56582,.
 :POKE56583,.
·30 POKE56589,130:POKE56590,17:POKE56591,
 81
·40 DATA 72,138,72,152,72,164,2,200,192,1
 6,208,2,160,0,132,2,185,28,192,141,32
·50 DATA 208,141,33,208,76,81,254
·60 END:REM COLORS
·80 DATA 6,14,3,5,13,7,10,8,2,0,0,0,0,0,0
 ,0
·100 PRINT"[CLEAR]":POKE56580,101:POKE565
 81,1:POK56582,2
·110 FORA=90TO109STEP.1:POKE56580,A:NEXT:
 FORA=109TO90STEP-.1:POKE56580,A:NEXT
```

The display can be customized in any number of ways. First of all, you can make the splits different sizes. Try these POKEs:

```
SUPERSMALL:
POKE 56580,152:POKE 56581,.:POKE 56582,.
SMALL:
POKE 56580,100:POKE 56581,1:POKE 56582,.
BIG:
POKE 56580,23:POKE 56581,2:POKE 56582,.
ECONOMY SIZED:
POKE 56580,101:POKE 56581,1:POKE 56582,2
```

You can achieve some really wild effects by making slight changes to the number you POKE to 56580. Type GOTO 100 after the hack has been enabled to see an example. Also, using supersmall splits will make further use of the computer unpredictable, and any use of peripherals when the hack is enabled won't work. Disable with STOP/RESTORE first.

The colors used are contained in line 80. Change these to make different displays. Always be sure there are exactly 16 numbers in this line, and always disable the hack before you rerun the program.

Here are some other POKEs you may wish to try:

```
BORDER ONLY
POKE 49172,32:POKE 49175,32
BACKGROUND ONLY
POKE 49172,33:POKE 49175,33
BACKGROUND AND BORDER:
POKE 49172,32:POKE 49175,33
```

LET'S FACE IT—MAIL ORDER HOUSES ARE *NOT* ALL THE SAME!

SOFTWARE SUPPORT INTERNATIONAL

CONVENIENCE—FAST SERVICE—RELIABILITY—SUPPORT

Abacus Software

128 Internals (book)	15.36
128 Tricks & Tips (book)	15.36
1571 Internals (book)	15.36
Basic Compiler 64	24.97
Basic Compiler 128	36.97
Becker Basic for Geos	33.97
Cad Pak 64	24.97
Cad Pak 128	36.97
Chart Pak 64	24.97
Chart Pak 128	24.97
Cobol 64	24.97
Cobol 128	36.97
CP/M for the C-128(book)	15.36
Super C Compiler 64	36.97
Super C Compiler 128	36.97
Super Pascal 64	36.97
Super Pascal 128	36.97
Geos Inside & Out Book	15.36
Geos Inside & Out Disk	11.76
Geos Tricks & Tips Book	13.20
Geos Tricks & Tips Disk	11.76

ACCESS Software Incorporated

Echelon w/Lipstick	24.97
LB & Tourn & Exec 1	12.97
Mach 5	21.97
Mach 128	30.97
Tenth Frame	24.97
World Class Leader Board	24.97
W/C L/B Fam/Courses 1	12.97
W/C L/B Fam/Courses 2	12.97
W/C L/B Fam/Courses 3	12.97

ACCOLADE

Apollo 18	18.97
Bubble Ghost	18.97
Card Sharks	18.97
Fast Break	Call
Fourth & Inches	18.97
Hardball	18.97
Miniputt	18.97
Plasmatron	9.97
Power at Sea	18.97
Rack-em	Call
Serve & Volley	Call
Test Drive	18.97
The Train	18.97

ACTIONSOFT

Thunder Chopper	18.97
Up Periscope	18.97

ACTIVISION ENTERTAINMENT SOFTWARE

Aliens	21.97
Blackjack Academy	24.97
Crossbow	21.97
Faery Tales	30.97
Fire Power	15.97
Gee Bee Air Rally	18.97
Last Ninja	21.97
Maniac Mansion	21.97
Might & Magic	24.97
Might & Magic Hints	10.97
Rampage	21.97
Romantic Encounters	24.97
Sky Travel	30.97

Berkeley Softworks

Desk Pak I/Font Pak I/Geodex I	31.97
Desk Pak Plus	18.97
Font Pack Plus	18.97
Geos 64	36.97
Geos 128	42.97
GEOSBASIC	Call
Geocalc 64	30.97
Geocalc 128	42.97
Geofile 64	30.97
Geofile 128	42.97
Geoprogrammer 64	42.97
Geoprogrammer 128	42.97
Geopublish 64	30.97
Geospell	18.97
Geowriter's Workshop 64	30.97
Geowriter's Workshop 128	42.97

BOX OFFICE

$100,000 Pyramid	9.95
Alf	9.95
High Rollers	9.95

Brøderbund Software

Arcade Game Const. Set	18.97
Bank Street Writer	30.97
Bank Street Filer	18.97
Bank Street Mailer	15.97
Bank Street Speller	15.97
Carmen Sandiego Europe	24.97
Carmen Sandiego USA	24.97
Carmen Sandiego World	21.97
Cauldron 1 & 2	18.97
Magnetron	15.97
Print Shop	27.97
Print Shop Companion	21.97
P/S Graphics Disk 1	15.97
P/S Graphics Disk 2	15.97
P/S Graphics Disk 3	15.97
P/S Holiday Graphics	15.97
Quest for Clues (book)	18.97
Superbike Challenge	12.97

CAPCOM

Ghosts & Goblins	18.97
Hat Trick	18.97
Minigolf	18.97
Side Arms	18.97
1942	18.97

CINEMAWARE

Defender of the Crown	21.97
King of Chicago	21.97
SDI	21.97
Sinbad	21.97
Rocket Ranger	21.97
Three Stooges	21.97
Warp Speed (Cart)	30.97

DATA EAST

Breakthru	18.97
Commando	12.97
Ikari Warriors	18.97
Karate Champ	12.97
Karnov	18.97
Kid Niki	18.97
Kung Fu Master	12.97
Platoon	18.97
Speed Buggy	18.97
Victory Road	18.97

DATASOFT

Alternate Reality City	19.36
City Hint Book	7.12
Alternate Reality Dungeon	25.48
Dungeon Hint Book	7.12
Battle Droidz	16.30
Global Commander	19.36
Hunt For Red October	25.48
Rubicon Alliance	13.24
Tobruk	19.36
Video Title Companion 2	13.24
Video Title Shop w/Comp 1	19.36

Digital Solutions Inc.

Dictionary Disk	9.97
Digital Superpak 2	60.97
Pocket Filer 2	36.97
Pocket Planner 2	36.97
Pocket Writer 2	36.97

ELECTRONIC ARTS

Alien Fires	19.36
Amnesia	25.48
Arctic Fox	21.40
Bard's Tale I	25.48
Bard's Tale I Hints	9.16
Bard's Tale II	25.48
Bard's Tale II Hints	9.16
Bard's Tale III	25.48
Bard's Tale III Hints	9.16
Chessmaster 2000	25.48
Chuck Yeager's AFS	22.42
Dan Dare	13.24
Delta Patrol	13.24
Demon Stalker	19.36
Earth Orbit Station	19.36
Instant Music	19.36
Legacy of the Ancients	19.36
Marble Madness	19.36
Master Ninja	19.36
Mavis Beacon Typing	25.48
Monopoly	19.36
PaperClip 3	31.60
PaperClip Publisher	31.30
Pegasus	19.36
Road Wars	19.36
Rockford	19.36
Scrabble	21.40
Scruples	25.48
Skate or Die	19.36
Skyfox II	19.36
Strike Fleet	19.36
Twilights Ransom	22.42
Twilights Ransom Hints	7.12
Wasteland	25.48
Wasteland Hints	9.16

EPYX

Arctic Antics	15.97
Battleship	Call
California Games	24.97
Champ. Wrestling	14.97
Create A Calendar	18.97
Death Sword	15.97
Destroyer	24.97
Fast Load Cartridge	24.97
Final Assault	24.97
4 X 4 Racing	24.97
The Games:Summer Edition	Call
The Games:Winter Edition	24.97
Graphic Scrapbook 1	15.97
Graphic Scrapbook 2	15.97
Graphic Scrapbook 3	15.97
Home Video Producer	30.97
Impossible Mission II	24.97
L.A. Crackdown	24.97
Metrocross	12.37
Mind Roll	24.97
Spiderbot	17.16
Sporting News Baseball	24.97
Sports-A-Roni	15.97
Street Cat	12.37
Street Sports:Baseball	24.97
Street Sports:Basketball	24.97
Street Sports:Soccer	24.97
Sub Battle Simulator	24.97
Summer Games II	13.92
Tower Toppler	24.97
Winter Games	12.97
World Games	24.97

FIREBIRD

Advanced Art Studio	24.97
Carrier Command	Call
Elite	11.97
Elite Hint Book (Leroy)	5.77
Guild of Thieves	24.97
Knight Orc	24.97
Jinxter	21.97
Star Glider	24.97
The Sentry	24.97
Tracker	24.97

GAMESTAR

Champ. Baseball	18.97
Champ. Basketball	21.97
GFL Football	21.97
Star Rank Boxing II	18.97
Boxing I/Tennis	18.97

INKWELL

Flexidraw 5.5	24.06
Flexifont	20.76
Graphics Integrator 2	20.76
Light Pen (184C)	40.57
Graphics Gallery 1	20.75
Graphics Gallery 2	20.75

KRACKER JAX

Elite V3.0	6.95
Elite V4.2	14.95
Geobusters V4	9.95
Hacker's Utility Kit	19.95

MICROPROSE

Airborne Ranger	21.97
F-15 Strike Eagle	21.97
Gunship	21.97
Pirates	24.97
Project Stealth Fighter	24.97
Red Storm Rising	24.97
Silent Service	21.97

MINDSCAPE INC.

720	21.97
Bad Street Brawler	18.97
Block Buster	18.97
Bop & Wrestle/Infiltrator 2	Call
Captain Blood	21.97
Clubhouse Sports	21.97
Crossword Magic	30.97
Defender of the Crown	21.97
Deja Vu	21.97
Eagles Nest/Infiltrator 1	Call
Gauntlet	21.97
Harrier Combat Simulator	18.97
Indoor Sports	18.97
Indy Jones Temple of Doom	21.97
Misl Soccer	21.97
Paperboy	21.97
Road Runner	21.97
SuperStar Ice Hockey	21.97
SuperStar Soccer	21.97
Uninvited	21.97
Xevious	18.97

ORIGIN

Autoduel	33.36
Moebius	26.88
Ogre	20.40
Ultima 1	26.88
Ultima 3	26.88
Ultima 4	39.84
Ultima 5	39.84

PROFESSIONAL
Professional Software Inc.

Fleet Filer 64/128	24.97
Fleet System 2 + +	36.97
Fleet System 4	48.97

SSI — STRATEGIC SIMULATIONS INC

Eternal Dagger	24.97
Gemstone Healer	18.97
Gemstone Warrior	9.97
Heroes of the Lance	19.36
Phantasie I	24.97
Phantasie II	24.97
Phantasie III	24.97
Pool of Radiance	25.48
Questron I	24.97
Questron II	24.97
Realms of Darkness	24.97
Rings of Ziffin	24.97
Shard of Spring	24.97
Wargame Construction Set	18.97
Wizard's Crown	24.97

SHAREDATA

Concentration	9.95
Family Feud	9.95
High Rollers	9.95
Jeopardy	9.95
Jeopard Jr.	9.95
Wheel Of Fortune	9.95
Wheel of Fortune 2	9.95
Card Sharks	9.95

SOLUTIONS UNLIMITED

Billboard Maker	24.97
Grafix Link	12.97
Icon Factory	24.97
Photo Finish	18.97
Screen F/X	21.97

SPRINGBOARD

Certificate Maker	26.88
Certificate Maker Libr.I	20.40
Newsroom	33.36
Newsroom Clipart 1	20.40
Newsroom Clipart 2	26.88
Newsroom Clipart 3	20.40
P/S Graphic Expander	23.64

subLOGIC

Flight Simulator II	33.97
Flight Sim Scenery 1	14.16
Flight Sim Scenery 2	14.16
Flight Sim Scenery 3	14.16
Flight Sim Scenery 4	14.16
Flight Sim Scenery 5	14.16
Flight Sim Scenery 6	14.16
Flight Sim Scenery 7	17.46
Flight Sim Scenery 11	17.46
Flight Sim Scenery 14	17.46
Flight Sim Scenery Japan	17.46
Flight Sim Scenery SanFran	17.46
Jet 2	27.36
Stealth Mission	33.96

TIMEWORKS

Accounts Payable	33.36
Accounts Receivable	33.36
Data Manager 2	17.16
Data Manager 128	33.36
Desktop Publisher	39.84
Electronic Checkbook	13.92
General Ledger	33.36
Inventory Management	33.36
Partner 64	33.36
Partner 128	39.84
Payroll Management	33.36
Swiftcalc/Sideways 64	17.16
Swiftcalc/Sideways 128	33.36
Sylvia Porter F/P 64	33.36
Sylvia Porter F/P 128	46.32
Word Writer 3	33.36
Word Writer 128	33.36

UNISON WORLD INCORPORATED

Art Gallery 1	15.97
Art Gallery 2	15.97
Art Gallery 4 Amer Hist	Call
Art Gallery 3	15.97
Printmaster Plus	21.97

MISCELLANEOUS

Art Worx/Bridge 5.0	21.97
Assembly for Kids (book)	11.76
Avalon Hill NBA Basketball	24.97
Basic 8	29.95
Beginner's Guide/Basic 8	19.95
Big Blue Reader 128/CP/M	27.97
Bob's Term Pro 64	30.97
Bob's Term Pro 128	36.97
Easy Working Triple Pak	12.97
B/W Business Form Shop	24.97
B/W Geos Word Publisher	24.97
CDA Graphic Transformers	21.97
Club Backgammon	20.40
CSM Drive Align Kit	30.12
CSM Protection Manual I	22.55
CSM Protection Manual II	26.15
Dark Castle (3-60)	21.97
Doodle	24.97
Flexfile 128	30.97
Font Master 2	30.97
Font Master 128	36.97
Geneology:Family Tree 64	36.97
Geneology:Family Tree 128	36.97
Geos Companion 1	18.97
Hes Mon 64 (cart)	9.95
IHT/CAD 3D	30.97
Konami:Contra	18.97
Konami:Rush'N Attack	18.97
Konami:Yie Ar-Kung Fu 2	18.97
Merlin Assembler 64	30.97
Merlin Assembler 128	42.97
Microlawyer	36.97
Microleague WWF Wrestling	24.95
Paul Whitehead Chess	21.97
Sam's/C-64 Prog. Base Ref. Guide	15.36
Softsync Pers. Newsletter	36.97
Strip Poker	18.97
Poker Data Disk 1	12.97
Poker Data Disk 2	12.97
Poker Data Disk 3	12.97
Star Empire	15.97
Superbase 64	48.97
Superbase 128	48.97
Superbase the Book	12.48
Super Sunday	23.64
Superscript 64	30.97
Superscript 128	42.97
Tab Books/Adv 128 Grap/Snd	12.48
Tab Books/C64 Trouble Shoot	13.20
Tab Books/C64 Serious Prog	8.88
Vampire's Empire	18.97
Wizardry/Sir Tech	24.97

ACCESSORIES

1541/71 Serial Cable	4.95
1541/71 Power Cable	4.95
Anti-Static Touch Strip	8.95
C-128 RGB Monitor Cable	5.95
C-64 Color Monitor Cord	4.95
C-64 Repairable Power Supply	39.95
RS-232 Interface	36.95
MW 350 Printer Interface	49.95
Aprospand 64	27.95
Aprospand Extender Cable	15.95
Aprotek 1200 Baud Minimodem	79.95
Commodore 1351 Mouse	37.95
Mouse Mat	8.95
1541 Dust Cover	8.95
1571 Dust Cover	8.95
C-64 Dust Cover	8.95
C-64C Dust Cover	8.95
C-128 Dust Cover	8.95
Disk Notcher-Square Cut	4.95
5.25" Disk Drive Cleaner	5.95
3.5" Disk Drive Cleaner	6.95
10 Cnt Disk Storage	1.95
50 Cnt Disk Storage	8.95
100 Cnt Disk Storage w/Lock	12.95
80 Cnt 3.5" Disk Stor. w/Lock	9.95
3.5" Disk Labels-25 cnt	1.00
5.25" Disk Labels-96 cnt	1.00
3.5" DS/DD Diskettes	1.29
5.25" DS/DD Diskettes-Black	.39
5.25" DS/DD Diskettes-10 colors	.79
Write Protects-100 cnt Black	1.00
Tyvek Sleeves-High Quality	ea .09
Slik Stik-Joystick	6.95
Quick Shot II-Joystick	7.95
Tac 2-Joystick	10.95
Tac 3-Joystick	11.95
Suncom Icon Troller	17.95
Surge Protector/Power Pad	39.95

MANY OTHER C-64/128 TITLES AVAILABLE — CALL FOR PRICES!

ALL NEW

ATTENTION C-128 OWNERS

Now that Commodore has released the C-128D with 64K of video RAM, we should be seeing 128 programs address this fantastic new feature soon.

BASIC 8 already has the capability of using all 64K of video RAM. If you own the C-128 in stock condition, you own all 16K of video RAM that Commodore felt was necessary. Using Basic 8 format and the full 64K of video RAM provides you with the ability to scroll through video memory as well as enhanced color resolution.

Up until now, to upgrade the C-128 to 64K of video RAM you would have to first search out the components, then find a competent repair outlet to desolder and install the parts. **What a hassle!**

SOLUTION — We have developed a module that simply plugs in to the mother board of your C-128. **No splattered solder — No heat damage — No hassle.**

This package includes full easy to follow installation instructions, a test program to validate proper installation and the plug-in upgrade module.

ONLY **$34.95**

AMIGA AMIGA AMIGA AMIGA

AMIGA SOFTWARE

4x4 Off Road Racing	24.97
Arkanoids	18.97
Awesom Arcade Pak	30.97
Barbarian	24.97
Bard's Tale	30.97
Bubble Ghost	24.97
California Games	24.97
Captain Blood	30.97
Deluxe Paint II	79.97
Digipaint	36.97
Digiview	134.97
Dive Bomber	24.97
Earl Weaver Baseball	30.97
Faery Tale	30.97
Fire Power	15.97
Flight Simulator II	33.97
Flight Sim Scenery #7	17.97
Flight Sim Scenery #11	17.97
FA/18 Interceptor	30.97
Heros of the Lance	25.97
Kindwords	60.97
Land of Legends	30.97
Leaderboard Golf	27.97
Leaderboard F/C #1	13.97
Maxiplan 500	109.95
Microfiche Filer	60.97
Money Mentor	59.97
Obliterator	24.97
Paladin	24.97
Phantasie	26.97
Phantasie III	26.97
Printmaster Plus	30.97
Printmaster Art Gallery #1	21.97
Printmaster Art Gallery #2	21.97
Printmaster Art Gallery #3	18.97
Printmaster Fonts/Borders	21.97
Road Wars	21.97
Rocket Ranger	30.97
Rockford	21.97
Sculpt 3D	66.97
Shadowgate	30.97
Strip Poker	24.97
Strip Poker Data Disk #4	13.97
Strip Poker Data Disk #5	13.97
Test Drive	27.97
The Director	42.97
Three Stooges	30.97
Time Bandit	24.97
Turbo	15.97
Uninvited	30.97
Vampire's Empire	21.97
Zoom	18.97

AMIGA AMIGA AMIGA AMIGA

ALL NEW

RENEGADE

ALL NEW

Project Phoenix. We assembled a team of the hottest archival programmers in the industry and gave them the challenge of their carreers: create a professional utility system that would set the tone for the future of Commodore personal computing - a system for the next decade and beyond.

Hundreds of expert-hours later, that team delivered to us a utility package of sobering power and scope. A package that could turn an average hacker into a superstar. A package created at the very boundary between what is and is not possible. A package called Renegade.

Renegade takes the very best ideas from the past and re-creates them, combining state of the art techniques with a conceptual grasp of the future of computing. The result is dynamic - classic utilities are transformed into muscular components of a system designed for speed, power, and flexibility.

Here are Some of the Features Built Into The RENEGADE!

★ Single or Dual High Speed Data Copier
★ Single or Dual State of the Art Nibbler
★ Directory Editor: Organize your Disks
★ GCR Editor: For the Experienced Hacker
★ Geos™ Modual: PARAMETERS and TOOLS
★ 200 Parms: For those tough to backup disks
★ New Parameter updates only $9.95 each
★ Compatable with C-64-128 with 1541/71 Drives
★ Single or Dual Ultra Fast File Copier
★ Capable of Archiving RapidLok Protection
★ Scrolling M/L Monitor with Drive Mon
★ Error Scanner with Unique Sector Editor
★ Byte Pattern Scanner: High Speed Searches
★ Upgradable Sub Menu: New tools in the works
★ Major RENEGADE upgrades only $9.95 each
★ Technical support available: of course

Renegade comes with over 250 parameters, and additional parameters are available every 2 months! Also, Renegade program updates are available to registered owners 3 times a year to ensure that Renegade will always be a step ahead of anything else on the market!

Now nothing can stop you from taking TOTAL control of your software! Renegade gives you the most advanced tools on the market for one remarkably low price. And, unlike some companies that claim to sell utilities unlimited in power, Renegade is produced by a company that knows that the customer is our most valuable asset.

Renegade: Next generation software - for the next generation hacker.

ATTENTION: REGISTERED RENEGADE OWNERS!
Parameter Module #2 is now available, Only $9.95

RENEGADE **Only $34.95**

GRAPHIC LABEL WIZARD

You will be delighted with this flexible and powerful tool. Without a doubt, the Graphic Label Wizard is the hottest label program around.

★ Print a graphic and up to 8 lines of text on a standard mailing label!
★ High Res display allows load and preview of up to four graphics!
★ Save created labels on disk for later recall, modification and printing!
★ Print catalogs of your Printmaster or Printshop compatable graphics!
★ Works with Epson compatible, and Commodore 1525, 801, 803 compatable printers!
★ Print text in any style your printer supports: Italic, Bold, Expanded!
★ Program disk includes exciting new graphics!
★ FREE BONUS: 100 NEW graphics created by Software Solutions!

A SUPER C-64 Utility from the people who brought you SuperCat!

ONLY $24.95

SYSRES™ ENHANCED

The best — we mean the Best Basic enhancement system for the C-64

★ Adds over 25 major commands to Basic
★ Extended Super DOS-Wedge ★ Scrolling thru Basic
★ Renumber, Trace, Search, and many other features
★ 1541/71 fast loader included
★ M/L monitor from Basic ★ Transferable to 1581

ONLY $39.95

SuperCat

Tired of searching endlessly through your disks to find the one title you're interested in? Frustrated by catalog programs that run out of memory or storage space every time your disk library grows?

Then now is the time to invest in SuperCat, the most sophisticated disk cataloging system available today for your Commodore C-64!

★ Catalog up to 640 disks and 5000 titles per disk!
★ Accepts disks with duplicate ID's
★ Reads titles from the directory of the disks to be cataloged.
★ Allows custom editing of titles being cataloged!
★ Prints a large variety of reports, even creates labels for your disks!
★ Operates with one or two 1541/1571 disk drives!

Find out why Ahoy! Magazine in the Feb. 1987 issue, gave high praises to this powerfull utility. SuperCat - A CLASSIC!

Only $24.95

C-128 CANNON

THE TOTAL COPY/UTILITIES PACKAGE CREATED JUST FOR C-128 OWNERS!

★ The Nibbler: Powerful! Works with single or dual 1541/1571 drives!
★ Fast Copier: For backing up data disks or for use with Kracker Jax!
★ File Copier: File transfers between 1541/1581/1571 drives!
★ 1581 Fast Copier: Copies from one 3.5" disk to another-1 or 2 drives!
★ MFM Copier: Copy unprotected IBM or CP/M formats-on 1571 drive!
★ Track & Sector Editor: 1541/1571 and even 1581 compatable!
★ Error Scanner: Full featured error scanner with on-screen display!
★ Density Scanner: Allows check for altered densities track by track!
★ Directory Editor: Reorganize the directories on your 5.25" disks!
★ Kracker Jax: 100 of our hottest, most popular parameters!
★ SPECIAL BONUS: Elite V3 for backups of Pocket 2 Series-FREE!

Your Commodore 128 deserves the best, so why not get the best!

ONLY $34.95

ALL NEW

Introducing the 1541 RAMBOard

Copy protection's "worst nightmare"

The dream has been there for years now: an inexpensive piece of hardware that, when added to your system, would allow you to backup almost every single piece of software ever released for the C64.

Software Support has just made the dream a reality!

We're proud to introduce the 1541 RAMBOard, a small card that can easily be installed into your 1541/1541C disk drive in 5 minutes using just a screwdriver. With this powerful card in place, backup hassles are a thing of the past! Working with special software, the RAMBOard will backup software that other utilities can't even scratch the surface of. And as new protection schemes arrive, we'll create new parameters to keep your RAMBOard operating behind enemy lines.

RAMBOard also comes bundled with a fast copier that can archive an unprotected data disk in under 50 seconds!

The concepts behind "card" systems are public domain. So why should you have to pay $44.95 or more for someone else's "card"? Software Support will sell you the hardware* for just $26.95!!! So if you don't have an unlimited income, don't worry. Just order your new RAMBOard from Software Support -- the company that's declared war on high prices.

RAMBOard -- Our Price: $26.95

Other companies "card" boards: $44.95 or more.

*The 1541 RAMBOard requires software to operate. This software can be either RAMBOard parameters, soon to be found on our Renegade program disk, or any of the other "card" software already on the market.

KRACKER KAX REVEALED I & II

Our knowledge of protection schemes has made us famous. Now find out how we do what we do best. Our books are your key to REAL knowledge. OTHERS ONLY GIVE SURFACE INFO - We dig deep, much deeper. Let us show you the ins and outs of todays copy protection. These books are a must for all serious Commodore users.

REVEALED BOOK I with Reset Button **$23.50**
REVEALED BOOK II with Hesmon Cartridge **$23.50**

REVEALED BOOK III COMING THIS FALL!!

FLASH! Kracker Jax Earns a 5 Star Rating In INFO's May-June '88 Issue

Kracker Jax is the powerful parameter based copying system that has taken the country by storm! What IS a parameter? It's a custom program that allows your 1541 or 1571 disk drive to strip ALL copy protection from your expensive software, leaving you with UNPROTECTED, TOTALLY BROKEN BACKUPS that can even be copied with a simple fast copier!

We declare **Kracker Jax** to be the best system of its kind on the market today! A bold claim? Maybe.

But don't take our word for it—if you want the REAL story on how good **Kracker Jax** is just ask one of our customers. Don't worry. You won't have any problem finding one.

Vols 1-2-3-4 **Only $9.95 ea.**
Vols 5-6-7 **Still Only $19.95 ea.**

SOFTWARE SUPPORT INTERNATIONAL

Reader Service No. 265

Ordering is simple: We accept money orders, certified checks, personal checks (of previous Software Support customers), VISA, M/C, Discover, and COD. Orders shipped to U.S.A., F.P.O., A.P.O., Canada, or Mexico, please add $3.00 per order for shipping and handling. COD available to U.S. customers only: add $2.25 additional per order. Foreign customers must call or write for exact shipping charges. Defective items are replaced at no charge IF and only IF you call for a Return Authorization Number. All in stock orders are processed within 24 hours. U.S. Shipping is by UPS ground in most cases. FAST 2nd DAY AIR available: add $1.00 per pound additional (U.S. 48 states only). U.S. software orders over 100 dollars will be shipped 2nd Day Air at our regular $3.00 S/H charge. Washington residents please add 7.5% additional for Sales Tax. All prices subject to change.

Program Submissions Invited
Need more info? Call or write for our free catalog.

Mail your order to: Software Support Int.
2700 NE Andresen Road / Vancouver, WA 98661

Or call our toll-free order line at 1-800-356-1179, 9am-5pm Pacific time Monday-Friday.
After hours orders accepted at (206) 695-9648 7 days a week.
Technical support available. Call (206) 695-9648, 9am-5pm Pacific time, Monday-Friday.

VISA · MasterCard · DISCOVER

DEALERS — WE HAVE THE SUPPORT YOU'RE LOOKING FOR!

```
SPRITE 1 AND 2 COLOR REGISTERS:
POKE 49172,39:POKE 49175,40
```

Feel free to try other VIC register numbers in the POKE statements. If you use the fourth example above, don't forget to put the sprites on the screen so you can see the effects!

Machine language programmers can learn a lot about timer A interrupts (*not* the same as an IRQ!) by examining the program. But whatever the case, I'm sure you'll enjoy the wild displays you can create. Have fun!

—Melvin Savage
Dear Park MD

TRACK TRACE

Have you ever wanted to find out just where that certain track and sector leads to? Well, *Track Trace* for the C-64 and C-128 is here to do the job.

Enter and save the program below. After running *Track Trace,* you will be prompted for a track and sector. Enter zero for both if you want to end the program. Otherwise, the program will proceed to trace tracks and sectors, starting with the one you entered.

Those of who who know how to examine track 18 (directory) can find the beginning track and sector of a program. This way, you can trace an entire program just to see how many sectors it takes up.

—Michael Jaecks
Alamogordo, NM

```
•10 REM TRACK TRACE
•15 PRINT"[CLEAR]"
•20 INPUT"ENTER TRACK,SECTOR (0,0 TO QUIT
 )";T,S:A=T:B=S:PRINT:D=0:IFT=0 THEN END
•30 PRINT"TRK:";A;"SEC:";B;:OPEN15,8,15:O
 PEN5,8,5,"#"
•40 PRINT"--> ";:D=D+1:IFD>1THENPRINTCHR$
 (13);:D=0
•50 PRINT#15,"U1";5;0;T;S:GET#5,A$:GET#5,
 B$:IFA$=""THENA=0:GOTO70
•60 A=ASC(A$)
•70 IFB$=""THENB=0:GOTO90
•80 B=ASC(B$)
•90 IFA>0THENPRINT"TRK:";A;"SEC:";B;:T=A:
 S=B:GOTO40
•100 PRINT"LAST SECTOR":PRINT
•110 CLOSE5:CLOSE15:GOTO20
```

ML LOCKUP LOCATOR

One of the most frustrating aspects of machine language programming is locating the exact section of code causing a machine lockup. Sometimes RUN STOP/RESTORE gets you out of trouble but you still have no clue as to why it happened. In BASIC, you have more than enough error messages that tell you what went wrong, but in ML you are on your own.

That's why I wrote the *Lockup Locator.* It installs a short ML wedge program at 53199 ($CFCF) and is armed by typing SYS 53199. After that, load and run your ML program with the lockup troubles. When your program reaches the lockup, just press the back arrow key and the system will reset itself. Then type SYS 53236 and the address that the computer was executing (or trying to execute!) at the point when you pressed the back arrow key will be displayed. Now you at least have a reference point to where your ML troubles are.

—Jeff Cobb
Otsego, MI

```
•10 REM LOCK UP LOCATOR
•20 FORX=53199TO53247:READA:B=B+A:POKEX,A
 :NEXT
•30 IFB<>6795THENPRINT"ERROR IN DATA. TRY
 AGAIN.":END
•40 PRINT"[CLEAR][DOWN] [4"*"] LOCK UP LO
 CATOR INSTALLED[3" "][4"*"]"
•50 PRINT" [4"*"] SYS 53199 TO ACTIVATE,[
 6" "][4"*"]"
•60 PRINT" [4"*"] PRESS '[BACKARROW]' TO
 TRIGGER, AND[3" "][4"*"]"
•70 PRINT" [4"*"] SYS 53236 FOR A REPORT.
 [5" "][4"*"]"
•80 DATA 120,169,207,141,21,3,169,220,141
 ,20,3,88,96,165,197,201,57,240,3,76
•90 DATA 49,234,104,104,104,104,104,141,2
 54,207,104,141,255,207,76,226,252,173
•100 DATA 255,207,174,254,207,32,205,189,
 96,0,0
```

BAM CHECK

Do any of your disks need to be validated? Why does a drive fail to validate some disks? Does your 1571 *really* need that new ROM upgrade? Find out with this program.

I didn't realize how *bad* my 1571 was until I started using this simple check program, which adds up the number of blocks of each directory entry. If the disk is sound, "Unused Blocks" should be equal to the "Blocks Free" total of the directory, which is read from the BAM. If "Unused Blocks" is bigger, then the disk needs to be validated. If it is smaller, then you're in trouble: your sloppy 1571 has been allocating blocks already used by other files, and your disk probably cannot be validated. Because this bug shows up when the 1571 is writing to the second side, you can avoid it by using single-sided disks. A better solution is to get the new ROM upgrade.

To use this program with a C-64, delete line 15 and "DIRECTORY" in line 140. With a 1581 (a very neat little drive), replace 664 with 3160 in line 20.

—Michael Laliberté
Outremont, QUE

```
•10 REM BAM CHECK
•15 IF RWINDOW(2)=80 THEN FAST
•20 DIM L%(300):TB=664:A=0:C=0:N=0:P=0:R=
 0:F=0:A$=""
•30 OPEN 15,8,15,"I0":OPEN 1,8,2, "$0,S,R
 "
•35 PRINT CHR$(147):PRINT" READING TRACK
 18 [3"."] >:COUNTED FILE":PRINT
•40 IF ST THEN CLOSE1:CLOSE15:GOTO 130
•50 IF P THEN N=N+1
•60 GET#1,A$:A=ASC(A$+CHR$(0)):IF C<254 T
```

Continued on page 44

COMMODARES

PROGRAMMING CHALLENGES

By Dale Rupert

Each month, we'll present several challenges designed to stimulate your synapses and toggle the bits in your cerebral random access memory. We invite you to send your solutions to:

Commodares, c/o *Ahoy!*
P.O. Box 723
Bethel, CT 06801

We will print and discuss the cleverest, simplest, shortest, most interesting and/or most unusual solutions. Be sure to identify the *name* and *number* of the problems you are solving. Put *your* name and address on the listings as well. Show sample runs if possible. Briefly describe your solutions and tell what makes them unique or interesting, if they are. You must enclose a stamped, self-addressed envelope if you want any of your materials returned. Solutions received by the middle of the month shown on the magazine cover are most likely to be discussed, but you may send solutions and comments any time. Your original programming problems, suggestions, and ideas are equally welcome. The best ones will become *Commodares!*

PROBLEM #58-1: UNDIVIDED PRIMES

This problem was submitted by Robert Boomers (Agnos, AR). Write a program which can find and display all prime integers (divisible only by themselves and 1) which are less than 30000. Only one catch: no division or multiplication is allowed. Sift through your programming experiences to solve this one.

PROBLEM #58-2: TIME ANGLE

Here's an interesting problem from Jim Speers (Niles, MI). The user enters a time in HH:MM:SS (hours, minutes, seconds) format. Write a program which gives the (smaller) angle between the hour hand and the minute hand in degrees. You need some acute thinking here.

PROBLEM #58-3: CHARITABLE TASK

This one from Necah Beyukdura (Ankara, Turkey) is certainly no giveaway. You have 9 donated envelopes of money to be distributed to 3 equally poor families. Each envelope contains at least $100 and less than $1000. Knowing the amount of money in each envelope, your task is to divide, in the most equitable manner, the 9 envelopes into 3 groups. (Unlike the "11 apples divided evenly by 12 people" problem, applesauce is not an acceptable answer.)

Stated more formally, write a program which takes 9 random integers from 100 to 999 and distributes them into 3 groups in such a manner that the sum of integers in each group is smaller or as close as possible to the sum of the 3 integers in any of the other groups.

PROBLEM #58-4: WORD ORDER

The user enters two words. The computer tells which of the two words would come first alphabetically if the letters of each word were first arranged alphabetically. To "COMMODARES" and "AHOY" the computer responds "COMMODARES" since "ACDEMMOORS" precedes "AHOY". Sort out your ideas before starting this easy one.

This month we will discuss the best solutions to *Commodares* from the April 1988 issue of *Ahoy! Problem #52-1: Later Date* was submitted by Wallace Leeker (Lemay, MO). The problem is to write a subroutine which determines whether the date stored in A$ is earlier than, later than, or the same as the date in B$. The dates are in MM/DD/YY format and are in the 20th century.

There were essentially two classes of solutions. One class converted the dates from strings into numbers, then compared them. A better method is simply to compare the strings directly. This program from Leslie Walden (Detroit, MI) is a good example of the preferred method:

```
•1 REM ==================================
•2 REM    COMMODARES PROBLEM #52-1 :
•3 REM        LATER DATE
•4 REM    SOLUTION BY
•5 REM        LESLIE WALDEN
•6 REM ==================================
•10 FOR N=1 TO 3: READ A$,B$
•20 GOSUB 50: NEXT N: END
•30 DATA 03/08/88, 09/16/74, 01/02/03, 01
/02/04
•40 DATA 08/10/80, 08/10/80
•50 AV$=RIGHT$(A$,2) + LEFT$(A$,2) + MID$
(A$,4,2)
•60 BV$=RIGHT$(B$,2) + LEFT$(B$,2) + MID$
(B$,4,2)
•70 IF AV$=BV$ THEN C$="THE SAME AS "
•80 IF AV$<BV$ THEN C$="EARLIER THAN "
•90 IF AV$>BV$ THEN C$="LATER THAN "
•100 PRINT A$; " IS "; C$; B$; "."
•110 RETURN
```

Lines 50 and 60 construct strings arranged in YYMMDD order. Now the strings may be compared to determine which date is larger. If YY is the same in both strings, MM will determine the order. If YY and MM are the same in the

two strings, DD decides which date is later. If the two strings are equal, so are the dates.

Jim Speers (Niles, MI) used this ON-GOTO structure to select the message to be printed:

```
70 ON 2+(AV$<BV$)-(BV$<AV$) GOTO 80,90,1
00
80 PRINT A$;" IS EARLIER THAN ";B$: RETU
RN
90 PRINT A$;" IS THE SAME AS ";B$: RETUR
N
100 PRINT A$;" IS LATER THAN ";B$: RETUR
N
```

If the logical expression (AV$<BV$) is true, its numeric value is −1. If the expression is false, its numeric value is 0. If neither logical expression is true, line 70 becomes ON 2 GOTO 80,90,100 which jumps to line 90, the second line number in the list. If the first logical expression is true, the second must be false, and line 70 becomes ON 1 GOTO 80,90,100 which branches to line 80. Similarly if the second logical expression is true, the minus sign in front of it causes line 70 to read ON 3 GOTO 80,90,100 which branches to line 100.

Congratulations also to John Desclin (Brussels, Belgium), Bob Miller (Memphis, TN), Terry Jernigan (Raleigh, NC), Dale Ickes (Uhrichsville, OH), and Frank Colaricci (Winter Park, FL), each of whom sent similar solutions using string comparisons.

Problem #52-2: Time Clock was suggested by Jim Speers. The challenge is to write a program which converts any number of minutes to hours and quarter hours (rounded to the nearest quarter hour). The shortest solution is this one from Claude Landusky (Honolulu, HI).

```
•1 REM ==================================
•2 REM    COMMODARES PROBLEM #52-2 :
•3 REM        TIME CLOCK
•4 REM    SOLUTION BY
•5 REM        CLAUDE LANDUSKY
•6 REM ==================================
•10 INPUT "ENTER MINUTES";M: PRINT INT((M
+7.5)/15)/4;"HOURS"
```

This solution prints hours and quarter-hours in decimal form. The problem didn't explicitly require fractions, although most other readers gave quarter-hours separate from the hours.

The following program from Jim Borden (Carlisle, PA) displays hours and fractional quarter-hours:

```
•1 REM ==================================
•2 REM    COMMODARES PROBLEM #52-2 :
•3 REM        TIME CLOCK
•4 REM    SOLUTION BY
•5 REM        JIM BORDEN
•6 REM ==================================
•10 INPUT "MINUTES PAY";MN: HR=INT(MN/15+
.5)/4: M=(HR-INT(HR))*4
•20 PRINT"PAYROLL TIME = "INT(HR);M"[LEFT
]/4 HOURS.": IF MN>0 GOTO 10
```

Notice that Jim's formula for rounding to the nearests quarter hour is equivalent to Claude's formula above.

Commodares readers can't resist a challenge, especially when it comes to beating someone else's program speed. Case in point is *Problem #52-3: Clock Segments* submitted by Bob Renaud (Washington, MA). Given a 24-hour, 6-digit, 7-segment digital clock, you are to write a program which displays in HH:MM:SS format all the times during the day when exactly half of the 42 digit segments are turned on. The program must print the number of times each hour and the total number of times within a day which meet this criterion. Times run from 00:00:00 to 23:59:59. The numbers of segments lit for numerals "0" through "9" are 6, 2, 5, 5, 4, 5, 5, 3, 7, and 5. Bob's program ran on the C-64 in 247 seconds.

This solution from Craig Ewert (Crystal Lake, IL) is interesting in the fact that it runs more slowly on the C-128 in FAST mode (220 seconds) than it does on the C-64 (212 seconds).

```
•2 TI$="[6"0"]":DIMCH(23):UL=2:PRINT"[CLE
AR]"
•3 DATA2,0,1,4,7,8:DATA2,3,5,6,9:DATA5,6,
2,4,3,7
•4 FORI=0TO10:READNM(I):NEXT:RESTORE:FORI
=0TO10:READP$(I):NEXT
•5 FORI=0TO5:READSA(I):NEXT
•10 FORX1=0TO2:C1=SA(X1)
•15 FORX2=0TOUL:C2=C1+SA(X2)
•20 FORX3=0TO3:C3=C2+SA(X3):IFC3>15THEN55
•25 FORX4=0TO5:C4=C3+SA(X4):IFC4>17THEN50
•30 FORX5=0TO3:C5=C4+SA(X5):IFC5>19THEN45
•35 FORX6=0TO5:C6=C5+SA(X6):IFC6=21THEN70
•40 NEXT
•45 NEXT
•50 NEXT
•55 NEXT
•60 NEXT
•65 UL=5:NEXT:GOTO200
•70 HN=NM(X1)*10:LA=X2:LB=X2:IFX2=0THENLA
=6:LB=10
•71 IFX2=0ANDX1=0THENLA=6:LB=7
•75 LC=X3:LD=X3:IFX3=0THENLC=6:LD=8
•80 LE=X4:LF=X4:IFX4=0THENLE=6:LF=10
```

```
85 LG=X5:LH=X5:IFX5=0THENLG=6:LH=8
90 LI=X6:LJ=X6:IFX6=0THENLI=6:LJ=10
95 FORY2=LATOLB:HC=HN+NM(Y2)
96 FORY3=LCTOLD:FORY4=LE TOLF
98 Y5=LG
99 Y6=LI
100 PRINT"[HOME]"P$(X1);P$(Y2);":";P$(Y3
);P$(Y4);":";P$(Y5);P$(Y6)
101 CH(HC)=CH(HC)+1:CD=CD+1
102 IFLJ>Y6THENY6=Y6+1:GOTO100
103 IFLH>Y5THENY5=Y5+1:GOTO99
105 NEXT:NEXT:NEXT:GOTO40
200 FORI=0TO23:PRINTI,CH(I):NEXT:PRINTCD
:PRINTTI$:END
300 REM =================================
310 REM   COMMODARES PROBLEM #52-3 :
320 REM      CLOCK SEGMENTS
330 REM   SOLUTION BY
340 REM      CRAIG EWERT
350 REM =================================
```

It is up to you to figure out how the program works. From Craig's notes, P$ stores the printable characters, and SA stores the number of segments. Since 2, 3, 5, 6, and 9 all light 5 segments, they are handled by special loops within the print routine in lines 70-105. Craig mentioned that Y5 and Y6 are not within loops because the C-64 runs out of stack space if they are. NM stores the actual digit represented by the indexes X1-X6 and Y2-Y6, used only for counting the occurrences for each hour in lines 70 and 95. Any clues as to why this runs so slowly on the C-128?

The shortest solution of reasonable speed is this one for the C-128 from Charles Kluepfel (Bloomfield, NJ).

```
1 REM ===================================
2 REM    COMMODARES PROBLEM #52-3 :
3 REM       CLOCK SEGMENTS
4 REM    SOLUTION BY
5 REM       CHARLES KLUEPFEL
6 REM ======= C-128 ONLY  ============
7 T=TI
10 DATA 6,2,5,5,4,5,5,3,7,5:FOR I=0 TO 9
: READ CT(I): NEXT
20 DIM C(59),TH(23):K=0:FOR J=0 TO 5:FOR
I=0 TO 9:C(K+I)=CT(J)+CT(I):NEXT:K=K+10:
NEXT
30 PUDEF"0":PRINT"[CLEAR]";:TH=3:FR=4:TW
=21
40 FOR H=0TO23:CH=C(H):FORM=0TO59:CM=TW-
C(M)-CH:IFCM>THTHENBEGIN:IFCM>FRTHENBEGI
N:FORS=0TO59:IFCM=C(S)THENPRINTUSING"##:
##:##";H,M,S;:PRINT"[HOME]";:TH(H)=TH(H)
+1
41 NEXT:BEND:ELSEPRINTUSING"##:##:##";H,
M,11;:PRINT"[HOME]";:TH(H)=TH(H)+1:BEND
45 NEXT:NEXT
47 PRINT"[DOWN]":FORI=0TO23:PRINTTH(I);:
TT=TT+TH(I):NEXT:PRINTTT
50 PRINT:PRINT(TI-T)/60
```

Charles makes fine use of that "formerly obscure" command PUDEF. The statement in line 30 defines the filler character to be a zero instead of a space in the PRINT USING instruction. This program runs on the C-128 in FAST mode in 228 seconds. C-64 users may enjoy converting this program into BASIC 2.0 and testing it.

Some other solutions to this problem were somewhat faster than these, but they were also much longer (using more sophisticated indexing) or incorrect (allowing duplications). Jim Speers mentioned one of his axioms that programming tradeoffs are speed v. memory requirements, and/or flexibility v. complexity. No argument here.

Let's wrap it up with *Problem #52-4: Factor Fun,* suggested by Dale Ickes (Uhrichsville, OH). The problem is to display all factors of an integer specified by the user. A goal is to do this as quickly as possible for large numbers.

Dale sent the following program, which he wrote to help his daughter.

```
1 REM ===================================
2 REM    COMMODARES PROBLEM #52-4 :
3 REM       FACTOR FUN
4 REM    SOLUTION BY
5 REM       DALE ICKES
6 REM ===================================
10 INPUT"FACTOR WHAT NUMBER";A
20 FOR C=A TO 1 STEP -1
30 B=A/C: IF B=INT(B) THEN PRINT B;
40 NEXT
50 PRINT: GOTO 10
```

Indeed it runs very slowly for larger numbers. One of the easiest ways to speed up this program is to check for divisors only as large as the square root of the given integer (the other factor will always be larger than the square root, but is found when dividing by the smaller factor). The next level of speed improvement for large integers is to use a table of prime numbers which has previously been created.

Claude Landusky sent this solution which incorporates both of these features.

```
1 REM ===================================
2 REM    COMMODARES PROBLEM #52-4 :
3 REM       FACTOR FUN
4 REM    SOLUTION BY
5 REM       CLAUDE LANDUSKY
6 REM ===================================
10 TI$="[6"0"]":PRINT"INITIALIZATION TAK
ES OVER AN HOUR FOR N TO REACH 31627[3".
"]":PRINT:PRINT"N ="
20 DIM P(3500):P(0)=2:C=1
30 FORN=3TO31627STEP2:X=0:PRINT N;
40 Q=P(X):IFQ*Q>NTHENP(C)=N:C=C+1:GOTO70
50 IFN/Q=INT(N/Q)THEN70
60 X=X+1:GOTO40
70 NEXT:PRINT TI$
80 INPUT"NUMBER TO FACTOR";N:TI$="[6"0"]
":F=0:C=0:IFN>999999999THEN80
90 Q=P(F):IFQ*Q>NTHEN120
```

```
•100 IFN/Q=INT(N/Q)THENN=N/Q:C=1:PRINTQ;:
GOTO100
•110 F=F+1:GOTO90
•120 IFC=0THENPRINT"PRIME"=;:GOTO140
•130 IFN<>1THENPRINTN;
•140 PRINT:PRINTTI$:GOTO80
```

The disadvantage of this program is that it takes over an hour to calculate prime numbers up to 31627 (the smallest prime greater than the square root of 999,999,999) and store them in P( ). Once they are calculated, the program quickly finds prime factors of integers up to 999,999,999. Do you know the prime factors of 987,654,321?

You might modify this program by removing the PRINT N; in line 30. Also change the limit in line 30 from 31627 to something smaller, thereby limiting the range of input values allowed. Using a limit of 300 in the FOR-NEXT loop lets you find factors of integers up to 300*300 or 90,000. You may also add instructions to this program to save the prime numbers in a separate disk file. The next time you run the program, just load the prime numbers from disk rather than calculate them. The time savings will be tremendous.

Notice that Claude's program gives only prime factors. You must change it in order to list all pairs of integer factors as given by Dale's program. For example, the prime factors of 12 are 2, 2, and 3. On the other hand, 12 can be factored as 1*12, 2*6, and 3*4. Changing this program to list all pairs of factors could be an interesting task in itself.

Hopefully you will have fun with this month's challenges. Keep those solutions and problems coming. □

Congratulations also to these readers who have not already been mentioned:

Saul Betesh (Kingston, ONT)
Carlton Burton (Easton, TX)
Stephen Byers (Monteagle, TN)
Tony Casciato (Virginia Beach, VA)
Carlos Centeno (Lares, PR)
David Dean (APO, NY)
Jerry Dellacca (Beech Grove, IN)
Jason Denlinger (Walkersville, MD)
Tim Farrell (Antigonish, NS)
Jesus Geliga-Torres (Aguadilla, PR)
Louis Lemire
Elizabeth Lindsey (East Brookfield, MA)
Larry Miranda (Toronto, ONT)
John Murphy (Neosho, MO)
William O'Keefe (Niagara Falls, ONT)
David Rice (Phoenix, AZ)
Bob Rispoli (Ridge, NY)
Andrew Rosenthal (Flushing, NY)
Herbert Schlickenmaier (Alexandria, VA)
Wayne Silberman (Thornhill, ONT)
D. Topper (Gibsons, BC)
Sarah Walden (Detroit, MI)
Terry Wollner (Tucson, AZ)
Patrick Ziegler (Bismarck, ND)

FLOTSAM

We welcome letters on all aspects of life in the Commodore universe. Send your comments (typed and double-spaced, please) to *Flotsam*, c/o *Ahoy!*, Ion International Inc., 45 West 34th Street—Suite 500, New York, NY 10001. We can only print a small sampling of the letters we receive, but we reply personally to as many as we can—and we read every one!

I am in the middle stages of publishing a C-128 user's guide to public domain, freeware, and shareware programs. I am of the belief that the 128 is a fine computer, and I would like to see it around a few more years.

Could you please let your readers know about this book? I need to hear from more 128 authors, and to get their software out to all interested users. I have reviewed over 200 fine C-128 programs for this book, but am still lacking in a few areas. I know your magazine is always cramped for space, but this help will not only benefit me, but all 128 users.

Here's what our book will have:

Reviews of all types of 128 software (no commercial programs); BBS's, wordpros, terms, databases, games, utilities, business software, and much more. Also, we will have articles from public domain writers, info on their latest products, and updates on their most current versions. We will supply addresses to all public domain services that wish to be in our book (at no charge). Also, we will be talking to people about hardware upgrades. We will also have a column on the inside front cover for people and companies that helped us get all the information we needed.

Thanks.

—Brad S. Burean
Blynd Dog Publishing
9410 E. 18th Terrace
Independence, MO 64052

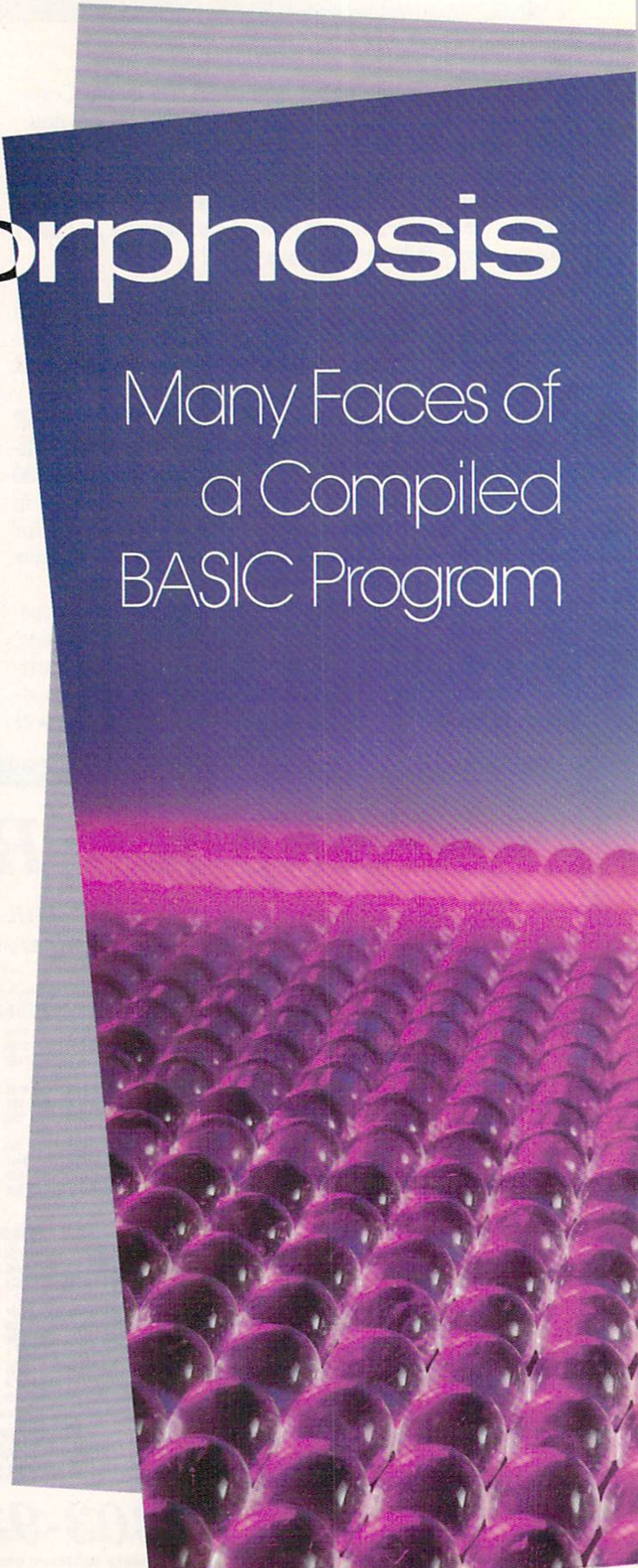
I must write to protest your past review of Epyx's *Sub Battle Simulator* (February '88 *Ahoy!*). From the review one gets the impression that the game is well-designed and worth purchasing. In reality, the simulation is incredibly bad and I cannot believe that a company with Epyx's impressive record would release such a half-baked piece of software. Your reviewers did not even point out that the simulation allows the player to cross land masses with the sub, as well as to operate indefinitely with a dead battery. Lack of attention to these details makes for very ridiculous and unconvincing gameplay (I consider these lapses to be bugs). What is the point of a "simulation" that does not follow obvious rules of logic?

Continued on page 48

RUPERT REPORT

BASIC Metamorphosis

Many Faces of a Compiled BASIC Program



PHOTOGRAPHY © CHRISTOPHER GOULD 1988

Take a rather drab, slow-moving BASIC program and compile it. Lo and behold – a sleek, fast machine language program appears. Last month we presented a bare-bones compiler written in BASIC to generate this metamorphosis. This month we will examine the details of the compilation process.

We will follow the compiler step by step. Hopefully when we are finished, you will be able to modify the compiler yourself and increase its capabilities. This is not a task for the novice. It will take some study, a knowledge of BASIC, and some machine language skills. If you have been looking for a challenging and rewarding software project, this may be it.

OVERVIEW

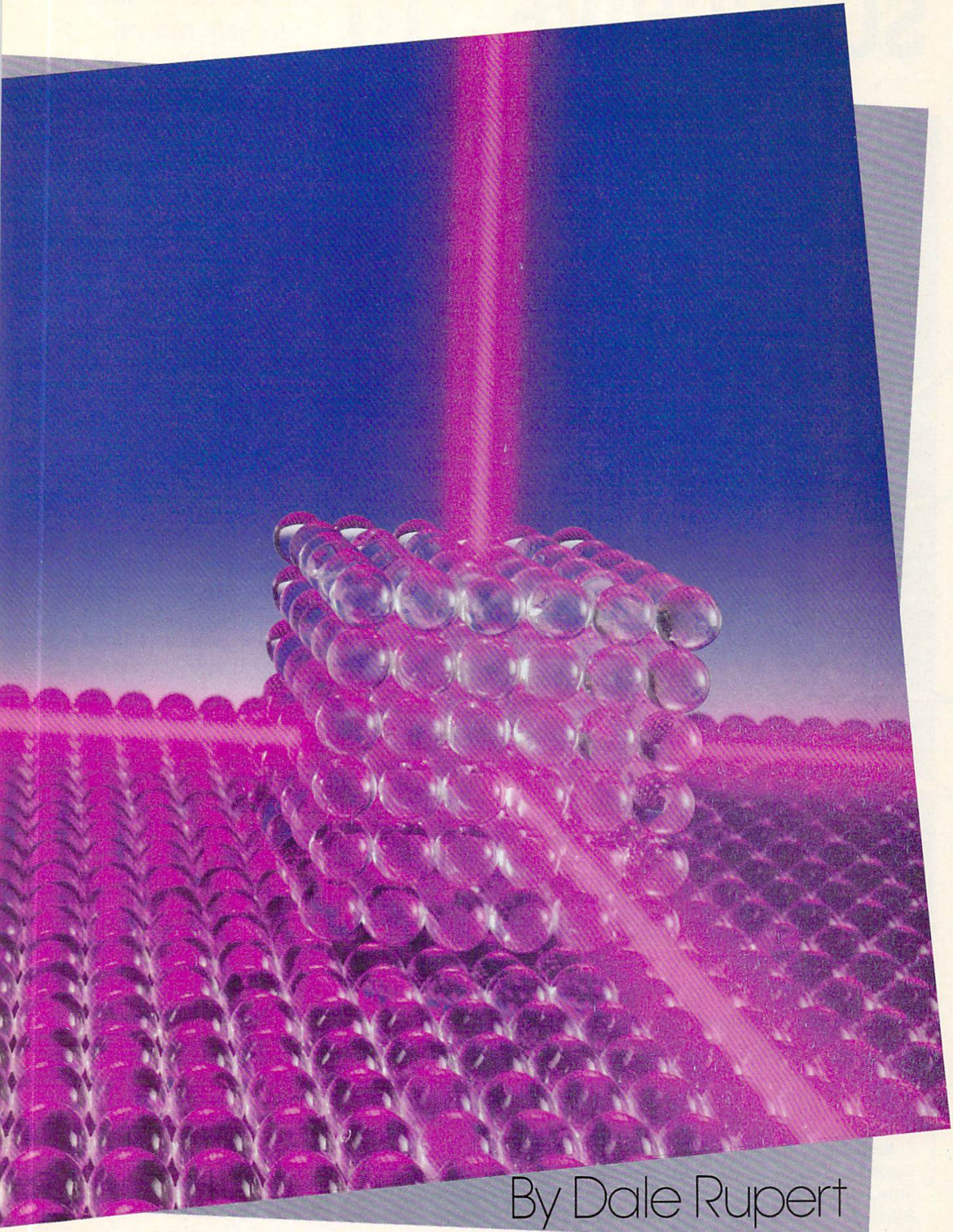
The compiler, called *Mini-comp*, is listed on page 62. It is nearly the same as the original version presented last month. Later we will discuss the few changes made to last month's program. We will also mention some changes required to make the compiler run on the C-128. First let's review the procedure for using *Mini-comp.*

With *Mini-comp* loaded into memory, add your BASIC program to be compiled (called the source code) between lines 0 and 999. Delete the lines of the sample program given in the listing. Line 999, or the last line of your program, should be an END statement. Debug your source code in BASIC as you normally do. RUN your program and edit it until it works the way it should. When the bugs are gone, compile it.

To compile your program, type RUN 2000. This bypasses the source code and jumps into the compiler. If the compilation is successful, you will see instructions to use the SYS command to execute the compiled program. Your BASIC program has been compiled (translated) into machine language and stored in memory starting at address 49408. The program in memory at that addresss is called the *object code.* The object code is an executable machine language program.

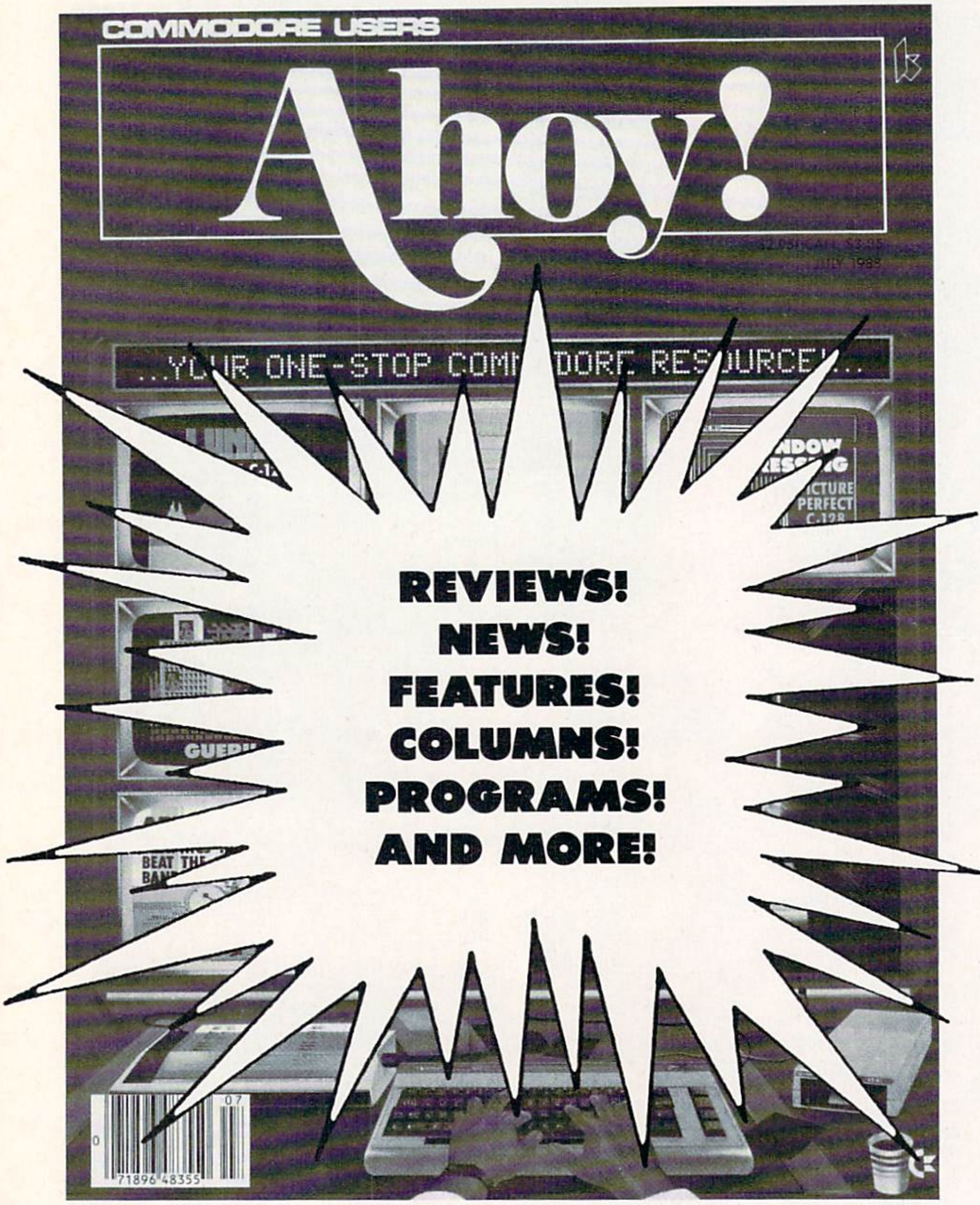
Type SYS 49408 in direct mode to run the compiled version of your program. The sample program given in the listing last month runs about 30 times faster when compiled than the corresponding interpreted program does.

If your compilation was not successful, the compiler describes the first error it found, and you will see a statement identifying the line number containing a syntax error. Recall from last month that the syntax of *Mini-comp* is very



By Dale Rupert

SUBSCRIBE TO



"You don't have to sell me on *Ahoy!* But why should I subscribe when I can pick it up at the newsstand?"

Here Are a Few Good Reasons:

1. You SAVE 33% off the newsstand price when you subscribe for two years (16 issues) for $31.95. That's under $2.00 an issue! Single year subscribers do almost as well—28% off the newsstand price. You'll do even better if you subscribe to the *Ahoy!* Disk Magazine—disk and magazine packaged together—over $30 off the yearly newsstand price!

2. You're certain never to miss an issue of your favorite Commodore magazine—crammed from cover to cover with C-64 and 128 news, reviews, articles, and programs! (But we forget—you know that already!)

3. You can enjoy the privileges of the *Ahoy!* Access Club, including monthly discounts from mail order advertisers and retailers around North America.

Convinced? Then fill out the postpaid card bound between pages 50 and 51 and return it today! (If someone beat you to it, use the coupon here.)

Enter my subscription to *Ahoy!*	**Enter my subscription to the *Ahoy! Disk Magazine***
☐ One Year (8 issues) $16.95 (Outside US $22.95)	☐ One Year (8 issues) $65.00 (Outside US $90.00)
☐ Two Years (16 issues) $31.95 (Outside US $42.95)	☐ Two Years (16 issues) $120.00 (Outside US $160.00)

☐ Payment enclosed: $__________ ☐ Please bill me.

☐ MasterCard ☐ VISA Card #____________________ Exp. Date__________

Signature______________________________

Name______________________________

Address______________________________

City____________________ State__________ Zip__________

Send coupon or facsimile to: *Ahoy!*, P.O. Box 341, Mt. Morris, IL 61054-9925

limited compared to standard BASIC. Only single-letter variable names are allowed. Only integer variables are supported (values from −32768 to 32767). The entire syntax is represented in the source line from 5 to 999 this month. These are all the allowed forms of instructions that *Mini-comp* can handle. Fix up any syntax errors, then re-compile the program with RUN 2000 until it is successful.

MEMORY MAP

On the C-64, RAM is available from $C000 to $CFFF for user programs. *Mini-comp* stores the object code in memory beginning at location PM which is 49408 ($C100 hex) in line 2060. Currently the space from 49408 to 53247 ($CFFF) can be used for your object code. EM in line 2070 stores the last available address for object code.

Variables are each stored in two bytes (least significant byte first) beginning at 49152 ($C000). The two bytes of variable A are stored in 49152 and 49153 ($C001). B's value is stored in 49154 and 49155, and so on, up to Z which is stored in locations 49202 and 49203. When the compiler variable C stores the ASCII value of a source code variable's name (A=65, B=66, ..., Z=90), this formula in line 1310 calulates the variable's storage address AD:

$$AD = (C-65)*2 + VM$$

VM is given the value 49152 in line 2050 as the start of variable storage. RAM from 49204 up to the start of object code at 49408 is unused.

The memory map for the C-64 is as follows:

```
Basic Source Code:         2049 - 40959 ($0801 - $9FFF)
Compiler Storage
  Integer Variables A-Z:  49152 - 49203 ($C000 - $C033)
  Unused:                 49204 - 49407 ($C034 - $C0FF)
  Object Code:            49408 - 53247 ($C100 - $CFFF)
```

TRANSLATION PROCESS

Last month we discussed the method by which the compiler parses the source code, fetching line numbers and instruction tokens. We also outlined the functions of the general purpose subroutines starting at line 1000. We discussed the array LL( ) which stores source code line numbers and the starting addresses of the corresponding object code. Finally we analyzed the operation of the "jump table" represented by array JT( ).

Now let's see how the compiler translates BASIC source code into machine language. We saw that there are six main paths of the syntax diagram corresponding to BASIC statements IF, GOTO, PRINT, REM, and END, as well as the assignment statement (an understood LET statement). Line 2220 branches to the routine at line 3000 if the first byte in the source line is a variable. Lines 2230 through 2270 handle the other types of statements.

Consider the source code statement

```
10 B=5
```

How is this translated into machine language? Clearly the purpose of this statement is to store the quantity 5 in the memory location assigned to the variable B. The compiler must convert the 5 into a two-byte integer quantity $0005 and then store those bytes in $C002 and $C003, B's storage address. The assembly language code to perform this operation is:

```
LDA #$05      ;get LSB of B
STA $C002     ;store it
LDA #$00      ;get MSB of B
STA $C003     ;store it
```

The first statement loads a 5 into the accumulator. This is the least significant byte (LSB) of variable B's value. The second statement puts this quantity into memory location $C002. Next B's most significant byte (MSB) which is 0 is loaded into the accumulator. Finally it is stored in location $C003.

Follow the steps of the compiler program as it translates the source code and stores the object code for 10 B=5. Assuming line 10 is the first program line, the source code in memory looks like this starting at address 2049:

Address:	value	;Meaning
2049:	9	;Next Line Pointer LSB
2050:	8	;Next Line Pointer MSB
2051:	10	;Current Line Number LSB
2052:	0	;Current Line Number MSB
2053:	66	;ASCII value of "B"
2054:	178	;BASIC token for "="
2055:	53	;ASCII value of "5"
2056:	0	;End of Line (EOL) token

The next line pointer and current line number are handled in lines 2130 through 2190 of the compiler. Line 2210 fetches and stores the next byte of source code (66) which is the ASCII value for "B".

Line 2220 calls the routine at line 1300 to see if the byte of source code in C is a variable. If the routine at 1300 returns with the variable flag VG true (it does in our example), the source code byte is a variable. The routine at 1300 returns with the two addresses for the variable's storage in A0, A1, A2, and A3. A0 and A1 are the LSB and the MSB of the address of the variable's LSB. A2 and A3 are the LSB and the MSB of the address of the variable's MSB.

In our example, the variable B is stored in $C002 (LSB) and $C003 (MSB). Consequently A0=2 ($02), A1=192 ($C0), A2=3 ($03), and A3=192 ($C0). Line 3010 transfers those values to D0, D1, D2, and D3 representing the "destination" address.

Line 3030 fetches the next byte of source code which is the token for "=". BASIC uses values greater than 127 for its key words. These values are called tokens. The token value for "=" is 178. Line 3030 generates an error if the value just fetched is not 178. According to the syntax diagram presented last month, "=" is the only token that may follow a variable. Likewise, the byte following the "=" may be either a minus sign, an integer, or a variable name. Line 3040 fetches the next byte, and line 3050 tests it.

For our example, the next byte is 53, the ASCII value for "5". Line 3060 stores a minus sign if there is one (token value 171). In our case, line 3065 stores a space character and then the character "5" in C$. This emulates the way BASIC handles numeric quantities for printing. Additional digits of the integer would be fetched and added a character at a time by lines 3070 through 3100 until the end of the program line is reached. The End of Line (EOL) character is always zero.

Now C$ stores the string equivalent of "5". It is converted into a numeric quantity in line 3120. Line 3130 verifies that the quantity is an integer (otherwise a BASIC error is generated). Line 3140 converts negative integers from −32768 to −1 into positive integers from 32768 to 65535. They are now in "two's complement" form. Line 3150 calls the routine at 1400 to break this number into its LSB and its MSB. Line 3170 stores those values in variables LSB and MSB.

Everything is ready to generate the object code for this program line. Line 3175 shows the assembly form of the code to be generated. It matches the assembly code listed above. The object code values are stored in C(1) through C(10). The object code for LDA immediate is 169 ($A9) (see your *Programmer's Reference Manual).* The object code for STA absolute is 141 ($8D). The other values of C( ) are the variables we have just discussed.

If "10 B=5" is the first line of our source code, CM has a value of 49152, the start of object code memory. The routine at 1500 POKEs the values of C( ) into memory starting at 49152. Those values for this example will now be:

Address	Decimal Value	Assembly Code
49152	169	LDA #(immediate)
49153	5	5
49154	141	STA (absolute)
49155	2	$02 (LSB)
49156	192	$C0 (MSB)
49157	169	LDA #(immediate)
49158	0	0
49159	141	STA (absolute)
49160	3	$03 (LSB)
49161	192	$C0 (MSB)

Consequently our source code in addresses 2049 through 2056 has become 10 bytes of object code shown above.

MANY FACES

We don't need such excruciating detail to understand the translation of the other paths of the syntax diagram. Figure 1 shows the many faces of a sample program and the process of its metamorphosis.

The first line of source code is the one we just discussed. You should be able to trace through the compiler program to see how the other BASIC statements are translated into the object code as shown in Part III of Figure 1. Notice that the assembler mnemonics are not actually generated. They are shown in the Figure to help us interpret the object code.

Line 30 is compiled into the integer addition routine as shown. The carry flag is cleared (CLC). Variable A's LSB is loaded into the accumulator, and B's LSB is added to it. The result is stored as C's LSB. The carry flag stores the carry, if there is one. Now A's MSB is loaded into the accumulator, and B's MSB and the carry are added. The result is stored as C's MSB.

Lines 3410 through 3580 of the compiler create the object code for the integer addition. Be careful not to confuse the variable names used in the remarks of the compiler with those of the source code in our example. The compiler remarks are for a source line A=B+C, and our example is C=A+B.

Line 40 of the source code is GOTO 30. This is compiled into a machine language JMP instruction. On the first pass, the compiler does not try to determine the address of line 30's object code to which to "jump." Instead it creates an entry in the Jump Table JT( ) indicating the current line number (40), the target line number (30), and the memory location after the JMP instruction where line 30's starting address will be stored.

Initially line 5120 stores 0's for the two bytes of line 30's starting address. In this example, the object code for line 30 starts at address 49430 and line 40 starts at address 49449. The object code for JMP (76) is stored in 49449. In pass 1 of the compiler, 0 is stored in 49450 and 49451.

During pass 2 of the compiler, lines 2420 through 2560 look up line 30's starting address in LL( ), convert it to LSB/MSB format, and store those bytes in addresses 49450 and 49451 after the JMP instruction. In our example, line 30's starting address is $C116 (49430). In LSB/MSB format this is $16/$C1 (22/193). So the object code for line 40 starting at address 49449 becomes 76, 22, 193.

CONDITIONAL BRANCHING

Mini-comp currently allows only the test IF A=B THEN mm where mm is a line number. It is easy also to allow IF A < B THEN mm. The routine at line 4000 handles the IF/THEN statement. Line 4040 tests for the "=" (token value 178) after the first variable name. To allow a "<" (token value 179), change line 4040, add 4045, and change just the first statement in line 4230:

```
4040 IF C<>178 AND C<>179 THEN STOP
4045 BR=144: IF C=178 THEN BR=208
4230 C(15)=BR: ...
```

Now if the conditional test of the IF/THEN statement is "A < B", line 4045 gives BR, the branch instruction, a value of 144. This is the op-code for BCC (Branch if Carry Clear). Otherwise BR has the value 208 which is the op-code for BNE (Branch if Not Equal).

The source statements

```
50 IF X<Y THEN 80
```

and

```
50 IF X=Y THEN 80
```

generate the following object code, shown in assembly language form:

```
1            LDA     X.MSB
2            CMP     Y.MSB
3            BNE     TEST2
4            LDA     X.LSB
5            CMP     Y.LSB
6   TEST2    ???     NEXT     ;BNE or BCC
7            JMP     mm
8   NEXT     ...
```

Lines 1 and 2 compare the MSBs of X and Y. If they are not equal, the carry flag tells whether X is greater than Y (carry clear) or whether X is less than Y (carry set). If they are equal, lines 4 and 5 are needed to test the LSBs. At line 6, the flags are set as follows:

Condition	Carry flag	Zero Flag
X<Y	set	clear
X=Y	clear	set
X>Y	clear	clear

If the IF/THEN statement is testing for X=Y, line 6 will contain a BNE (branch if zero flag not set) statement to bypass the JMP instruction in line 7. If the IF/THEN test is for X<Y, line 6 will contain a BCC (branch if carry clear) statement to bypass the JMP. In line 7, mm corresponds to the address of line 80 in object memory. It is determined just like the GOTO mm statement which we discussed earlier.

FIGURE 1
The Many Faces of a Compiled BASIC Program

I. Source Code on the Screen

```
10 B=5
20 A=B
30 C=A+B
40 GOTO 30
```

II. Source Code in Memory

decimal address	hex address	hex (lsb msb) next line ptr	decimal line #	decimal tokenized code and EOL
2049	$0801	$09 $08	10 00	66 178 53 0
2057	$0809	$11 $08	20 00	65 178 66 0
2065	$0811	$1B $08	30 00	67 178 65 170 66 0
2074	$081B	$24 $08	40 00	137 32 51 48 0
2082	$0824	$00 $00	–	

III. Compiled Assembly Code and Object Code

line BASIC	Assembler mnemonics	Hexadecimal object code	Decimal object code
10 B=5	LDA #5	A9 05	169 5
	STA B.LSB	8D 02 C0	141 2 192
	LDA #0	A9 00	169 0
	STA B.MSB	8D 03 C0	141 3 192
20 A=B	LDA B.LSB	AD 02 C0	173 2 192
	STA A.LSB	8D 00 C0	141 0 192
	LDA B.MSB	AD 03 C0	173 3 192
	STA A.MSB	8D 01 C0	141 1 192
30 C=A+B	CLC	18	24
	LDA A.LSB	AD 00 C0	173 0 192
	ADC B.LSB	6D 02 C0	109 2 192
	STA C.LSB	8D 04 C0	141 4 192
	LDA A.MSB	AD 01 C0	173 1 192
	ADC B.MSB	6D 03 C0	109 3 192
	STA C.MSB	8D 05 C0	141 5 192
40 GOTO 30	JMP (LINE 30)	4C ?? ??	76 ?? ??

VARIABLE STORAGE

A	LSB	$C000
	MSB	$C001
B	LSB	$C002
	MSB	$C003
C	LSB	$C004
	MSB	$C005
	...	

REM, END, AND PRINT

REM statements are handled by the routine at line 8000 which simply fetches the bytes of source code and ignores them. The END statement is converted in line 7000 into a machine language RTS (Return from Subroutine) instruction. This transfers control from your machine language object code back to the BASIC interpreter.

The only other instruction currently allowed by *Mini-comp* is the PRINT statement. Its variations are PRINT, PRINT A, or PRINT CHR$(A), each with an optional semicolon at the end. The routine at line 6000 compiles the PRINT statements.

PRINT and PRINT CHR$(A) both use the subroutine at line 6200 to print a single character on the screen. This subroutine calls the Kernal routine CHROUT at address $FFD2. CHROUT requires that the accumulator contain the ASCII value of the character to be displayed. Subsequently JSR $FFD2 displays that character.

PRINT without an argument is the same as "PRINT CHR$(13);" which prints a single carriage return. Line 6020 performs that task. PRINT CHR$(A) is handled at line 6300. The LSB of the variable A is loaded into the accumulator. Due to the fact that CHR$( ) only allows arguments from 0 to 255, the MSB of A is ignored.

The object code for PRINT

CHR$(A); is as follows:

```
LDA A.LSB
JSR $FFD2
```

The object code for PRINT is

```
LDA #13
JSR $FFD2
```

Line 6400 tests for a semicolon after PRINT CHR$(A). If there is no semicolon, line 6390 adds the object code to print a carriage return.

The most complicated source code statement to implement in this compiler is PRINT A. It is processed beginning at line 6040. The remark in line 6050 is corrected from last month's listing. The object code is simply this:

```
LDX A.LSB    ;Put A's LSB in X
LDY A.MSB    ;Put A's MSB in Y
JSR $C0E0    ;call Print Integer routine
```

The routine at $C0E0 is disassembled in Figure 2. The routine is called with A's MSB in the Y register and the LSB in the X register. After printing the leading space or minus sign, the routine converts A from a signed integer (−32768 to 32767) into an unsigned integer (0 to 65535). The C-64 BASIC ROM routine at $BDCD converts a 2-byte unsigned integer into a floating point quantity which is then converted into an ASCII string and printed.

This is a very roundabout way to implement PRINT A.

FIGURE 2

```
;====================================
;
;       RUPERT REPORT #58
;    PRINT N ROUTINE WHERE N IS
;  A SIGNED INTEGER -32768 TO 32767
;
;========= C-64 ONLY =============
$C0E0 A9 20       LDA #$20   ; " "
$C0E2 C8          INY        ;MSB
$C0E3 88          DEY        ;SET FLAGS
$C0E6 A9 2D       LDA #$2D   ;NO, "-"
$C0E8 20 D2 FF    JSR $FFD2  ;PRINT IT
$C0EB 98          TYA        ;GET MSB
$C0EC 10 0C       BPL $C0FA  ; >=0?
$C0EE 8A          TXA        ;NO, LSB
$C0EF 49 FF       EOR #$FF   ;1'S COMP
$C0F1 18          CLC        ;+1 GIVES
$C0F2 69 01       ADC #$01   ;2'S COMP
$C0F4 AA          TAX        ;SAVE IT
$C0F5 98          TYA        ;GET MSB
$C0F6 49 FF       EOR #$FF   ;1'S COMP
$C0F8 69 00       ADC #$00   ;+ CARRY
$C0FA 20 CD BD    JSR $BDCD  ;PRINT IT
$C0FD 60          RTS        ;DONE
```

It is also relatively slow during execution. You could write a different PRINT A routine and replace the DATA statements starting at line 11080 with your code. Your routine must convert the 2-byte signed integer in the X (LSB) and Y (MSB) registers into ASCII characters and print them. Your routine should start at $C0E0 and end with an RTS instruction.

C-128 CONVERSION

If you want to convert *Mini-comp* for the C-128, here are some suggestions. You must supply a PRINT A routine as described above. The routine at $BDCD for the C-64 is not valid for the C-128. There are some documented routines which you might use, however. The Kernal routine CHROUT (called BSOUT for the C-128) still resides at $FFD2 and works the same as on the C-64 to print a single character.

JSR $AF03 (GIVAYF) converts a 2-byte signed integer (MSB in accumulator, LSB in X register) into a floating point value in the floating point accumulator (FAC1). JSR $AF06 converts the value in FAC1 into an ASCII string at $0100 terminated with a 0 byte. Your routine could take characters starting at $0100 and print them using BSOUT until the zero byte is reached.

You must change the memory values in the following lines for the C-128. Here are some suggested values:

```
2050 VM=3072   :REM VARIABLES @ $0C00
2060 PM=3122   :REM OBJECT CODE @ $0C32
2070 EM=3583   :REM OBJECT CODE @ $0DFF
2080 BT=7169   :REM BASIC TEXT @ $1C01
```

Also change line 11010 to the starting address of your PRINT A routine and change the checksum value CS in line 11020.

A "pause" key feature has been added to *Mini-comp* since last month in lines 2300 and 2310. Press any key to stop the display after the current line. Press another key to continue. Also line 2595 has been added indicating the range of addresses occupied by the object code. Don't forget the change to the remark in line 6050.

Hopefully you now have an understanding of the compilation process. Following our discussion, you should be able to modify *Mini-comp* to handle other BASIC commands. For example, GOSUB can be modeled after the GOTO statement. Use JSR instead of JMP in the object code. The RETURN statement becomes simply RTS. It should not be difficult to save your compiled programs to disk for use by other programs. In future columns we will implement floating-point math operations, string functions, and additional BASIC statements.

If you are not interested in expanding *Mini-comp*, you might enjoy creating models using *Mini-comp's* limited syntax to implement more sophisticated BASIC statements and structures. FOR-STEP-NEXT, ON-GOTO, DO-WHILE can all be simulated with simpler IF-THEN-GOTO statements. Reduced instruction set (RISC) microprocessors are currently the trend for new hardware designs. This compiler gives you ample opportunity to work with a reduced instruction set. Enjoy it. □

SEE PROGRAM LISTING ON PAGE 62

BOMBS AWAY!

For the C-64

By John Fraleigh

Bombs Away! seems like a very simple game. A mad arsonist prowls the top of a building, armed with bombs and dynamite sticks. You are equipped with buckets of water which you must use to catch and extinguish the bombs. As I said, it seems very simple—but before you know it the bombs are dropping at a rate of about 15 per second.

You start the game with three buckets of water. You move them back and forth via a joystick plugged into Port 2. If you miss a bomb, it will of course explode, and it will cost you one of your buckets. Each consecutive level features more bombs and faster action. Also, the higher the level, the more points each bomb is worth. The game ends when you've lost all three buckets. A high score is kept. You'll need to be very quick to do well—good luck.

Entering the program requires the use of *Flankspeed* (see page 61). The starting address is $C000 and the ending address is $C55F. To start, SYS 49152.

The program takes advantage of the sprite, redefined character, color, and sound capabilities of the 64. It also wedges a short routine into the hardware interrupt to allow for more than two sprite multicolors and to divide the screen into two different screen colors. □

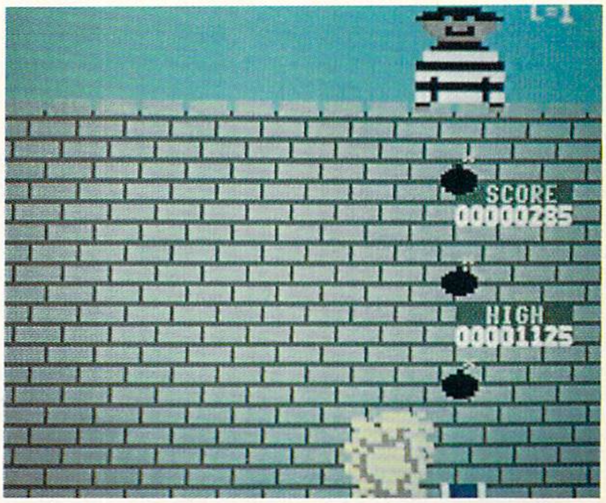


PHOTO: MICHAEL R. DAVILA

SEE PROGRAM LISTING ON PAGE 69

SIDE BORDER SCROLL EDITOR

Entering Your Screen's Forbidden Territories

By Michael A. Grotton

Over the many years of the Commodore 64's existence, many impressive products have come out for it. But every now and then there will appear a quirky little routine that is truly astonishing. The use of mathematical functions in sprite movements, for instance, was a ground-breaking achievement that originated in Europe and made its way to the US in due time. Now I'd like to share one of my own discoveries with you.

While traveling through Europe in the summer of '87 I stopped off at a little software shop in Holland. On one of the monitors I saw a picture setup that was occupying the entire screen. More than just the normal visible viewing area—the whole screen. The side borders, top border, bottom border—everything!

When I asked the store owner if I could possibly borrow an ML monitor and a computer for awhile, he rudely responded: "If you want to do that, go back to America and figure it out for yourself. I ain't gonna help you." It seems that European programmers are not too willing to share their secrets. Feeling a little cocky, I replied: "No problem!" And some months later, here I am.

Before getting into specifics, let's take a look at the editor I have written for you as an example of side border elimination. *Side Border Scroll Editor* is 10 blocks of code (about 2½K). It is actually two separate programs combined: the editor for editing the text, color, and speed of color changes, and the sprite scroll/side border eliminator. After entering the editor with *Flankspeed* (page 61), load it up and type SYS 49152 on a blank line and hit RETURN. After a moment or two you will see sprites entering a portion of the side border. For a better effect, set the border color the same as the screen color. I find that POKE 53281, 0:POKE 53280,0 works best.

When you are done viewing the scroll, type SYS 49541. This is the disabling SYS for the routine. Now type SYS 50688. This will enter the editor. Your options will be to:

1) Edit text
2) Edit preferences
3) Save the scroll
4) Quit the editor

If you choose to edit text, the screen will clear and a cursor will appear. Be careful not to scroll the screen, as what is on the screen will be transported to the sprite scroll area. Type your text, ending with 20 spaces, place the cursor at the end of your typing (or after the 20 spaces), and hit f1. The editor will place an endmark on the screen where your cursor was and move your text into the scroll's buffer.

Now for preferences. The sprite scroll has seven color phases which it cycles through to make the scroll flash. To change any one of the seven colors, move the arrow (with the cursor right/left key) below the color you want to change, and hit "+" to advance the color by one and "–" to decrease it by one. If you prefer a solid color scroll, simply turn each of the numbers to the designated color. Next the program will ask you for the speed. For this enter 1 for fast, 2 for medium, and 3 for slow. These control how fast the flashing will be. Your next two options are self-explanatory: Save to save the scroll as a separate routine, and Quit to exit the editor.

The only restriction on the scroll routine is that you cannot use memory between $2000 and $2300. This is used for data storage by the sprite scroll. Also, the scroll routine uses the hardware interrupt vector at $0314 and $0315, so not all DOS wedges/cartridges will work with it unless they offer "transparent" operation.

Now that you know how to create a scroll with side border sprites, here is a little info on how I eliminated the side border to allow the sprites to show their faces there.

For one thing, my routine only works if all sprites are on and are in the general vicinity of one another (i.e., the scroll sets up all 8 sprites next to each other on the same line). The most important factor is timing. I have two different ways of making sure everything happens when I want it to. One is a raster interrupt (check out *Mapping the 64* under $D01A). I set it to occur when the raster line reaches the same starting line as the sprites. Next I must make sure my vital code occurs at that exact moment. I accomplish this by using both a countdown routine and NOPs (No Operations). The NOPs take up about 2 cycles. The countdown routine works like this:

```
        LDY #$06
LOOP    DEY
        BPL LOOP
```

So it simply stalls the computer temporarily. Next I need the proper amount of NOPs to get the raster right where I want it. Now the trick: by toggling 53270 ($D016) in and out of 38 column mode (i.e., shrinking the border), the 64 will forget about drawing any border at all. Of course another part of it is necessary. By pushing the screen down using the vertical scroll register at 53265 ($D011) I insure that nothing will get inside the sprite area to mess up the speed of the raster. One interesting note is that by eliminating that part of the screen we get some garbage in its place. This is somehow controlled by the last viewed byte in the VIC chip's current bank. In the normal setting this byte lies at 16383 ($3FFF). By turning it to a zero the garbage is more than taken care of.

I hope you enjoyed this article/program. If you would like more info on the trick or a copy of my own source files, write me care of *Ahoy!* □

SEE PROGRAM LISTING ON PAGE 71

Ahoy! DISK

Why type in the listings in this month's *Ahoy!* when we've done it for you? All the programs in this issue are available on the current *Ahoy! Disk* for $8.95. Isn't your time worth more than that?

Use the coupon at the bottom of this page to order disks for individual months, a disk subscription, or the special anthology disks described below. (You can also subscribe to the *Ahoy! Disk Magazine*— disk and magazine packaged together at a special reduced rate. See the card bound between pages 50 and 51.)

BEST OF '84
Address Book (3/84)
Air Assault (5/84)
Alpiner (6/84)
Checklist (7/84)
Construction Co. (12/84)
DOS (9/84)
Emerald Elephant (10/84)
I Got Biorhythm (5/84)
Lawn Job (10/84)
Lunar Lander (4/84)
Math Master (5/84)
Micro Minder (8/84)
Multi Draw 64 (1/84)
Name That Star (4/84)
Post Time (6/84)
Salvage Diver (9/84)
Space Lanes (VIC) (3/84)
Space Lanes (64) (3/84)
Space Patrol (12/84)
VIC 40 (10/84)

BEST OF '85
Ahoy! Dock (9/85)
Auto Line Numbers (3/85)
Directory Manipulator (12/85)
File Lock (9/85)
Fontasia (2/85)
Futurewar! (2/85)
Gameloader (12/85)
Home Budget (3/85)
Jewel Quest (12/85)
Lightning Loader (11/85)
Midprint (1/85)
Music Tutor (1/85)
Salmon Run (3/85)
Screen Dump (8/85)
Telelink (2/85)
Tile Time (5/85)

BEST OF '86
Disk Cataloger (8/86)
Dogcatcher (9/86)
Fidgits' Music School (7/86)
Flap! (8/86)
Guardian (11/86)
Head to Head (2/86)
Highlight (8/86)
JCALC (4/86)
Meteor Run (8/86)
Mine Canyon (9/86)
Mountaineer Mack (10/86)
Star Strike (6/86)
Step On It (8/86)
Syntax Patrol (9/86)
Teleporter (11/86)
Tenpins (11/86)
1541 Alignment Check (2/86)

BEST UTILITIES
Auto Append (7/85)
BASIC Trace (4/85)
Base Conversions (9/84)
Booter (4/85)
Commodore Font Editor (5/85)
DOS (9/84)
Fastnew (9/85)
Formatter (8/85)
Lucky Lottery (7/85)
Maestro (10/85)
Micro Minder (8/84)
Printat (7/85)
Screen Bit Dump (6/85)
Screen Magic (6/85)
Start & End File Address (9/85)
Two Column Directory (5/85)
Windows (8/85)
1541 Cleaning Utility (10/85)

BEST UTILITIES II
Alarm Clock (1/86)
BASIC Relocator (1/86)
Bigprint (5/86)
Chrono-Wedge (4/86)
Disk Checkup (10/86)
Failsafe (5/86)
Long Lines (11/86)
Programmer's Aid (3/86)
Quick Change (10/86)
Reversed Remarks (7/86)
Screen Sleuth (7/86)
Screen Window (4/86)
Speech64 (9/86)
The Editor (12/86)
Trackdown (6/86)
Very-Able (3/86)
128 to 64 Autoboot (10/86)

BEST GAMES
Alpiner (6/84)
Construction Co. (12/84)
Disintegrator (8/85)
Dragon Type (9/85)
Duck Shoot (6/85)
Emerald Elephant (10/84)
Gators N Snakes (8/85)
Gravinauts (10/85)
Lawn Job (10/84)
Mystery at Mycroft Mews (10/85)
Name That Star (4/84)
Post Time (6/84)
Salvage Diver (9/84)
Script Analysis (7/85)
Space Patrol (12/84)
Tile Time (5/85)
Wizard of Im (7/85)

BEST GAMES II
Actionauts (4/86)
Air Rescue (4/86)
Brickbusters (3/86)
Crabfight (2/86)
English Darts (5/86)
Got a Match? (6/86)
Haunted Castle (1/86)
Head On (7/86)
Knockout (1/86)
Micro City (12/86)
Minotaur Maze (12/86)
Penguins (10/86)
Ski Folly (5/86)
Star Search (5/86)
Swoop (4/86)
The Last Ninja (9/86)

BEST C-128
Ahoy!Term (3/86)
Dark Fortress (1/87)
Dual Dump (9/86)
Extended Background Mode (4/87)
Infoflow (4/87)
List Formatter (4/87)
More BASIC 128 (2/87)
Nebergall Run (7/86)
Planet Duel (4/87)
Queen's and Bishop's Tours (6/86)
Rescue 128 (3/87)
The Artist (12/86)
Variable Manager (9/86)
Wordcount (7/86)
128 to 64 Autoboot (10/86)

We've collected Ahoy! 's best programs onto the Anthology Disks described here. Please note that many of these programs will be unusable without the documentation printed in the issues of Ahoy! listed in parentheses.

SINGLE ISSUE AND SUBSCRIPTION PRICES

(Postage and Handling Included)	U.S.A.	CANADA	ELSEWHERE
❑ October '88 *Ahoy!* Disk	$ 8.95	$ 12.95	$ 13.95
❑ 12-Month Disk Subscription	$ 79.95	$ 99.95	$124.95
❑ 24-Month Disk Subscription	$149.95	$179.95	—

BACK ISSUES $8.95

In Canada add $2.00 per issue; outside US and Canada add $4.00 per issue.

❑ ___	$8.95	❑ ___	$8.95
❑ ___	$8.95	❑ ___	$8.95
❑ ___	$8.95	❑ ___	$8.95
❑ ___	$8.95	❑ ___	$8.95

ANTHOLOGY DISKS $10.95 EACH

❑ Best of '84	❑ Best Utilities
❑ Best of '85	❑ Best Utilities II
❑ Best of '86	❑ Best Games
❑ Best of C-128	❑ Best Games II

Send coupon or facsimile to: *Ahoy!* Program Disk
Ion International Inc.
45 West 34th Street, Suite 500
New York, NY 10001

MC and VISA Accepted Call 718-979-9330

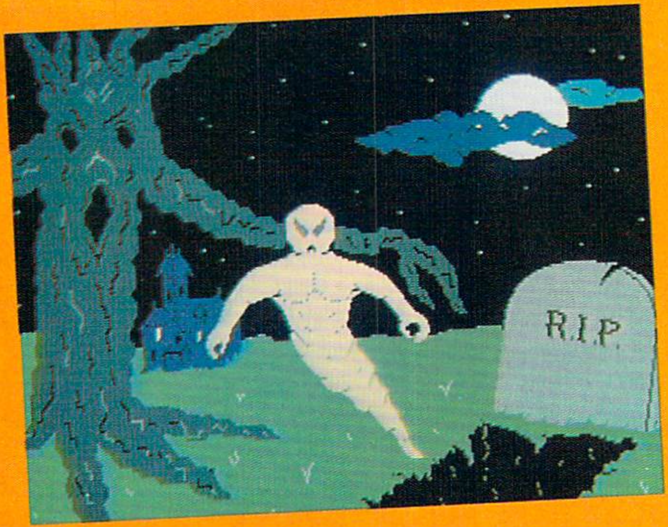
NAME ____________________

ADDRESS ____________________

CITY ____________ STATE ________ ZIP ________

Make check or money order payable to **Double L Software**. N.Y. State Residents must add 8¼% sales tax.

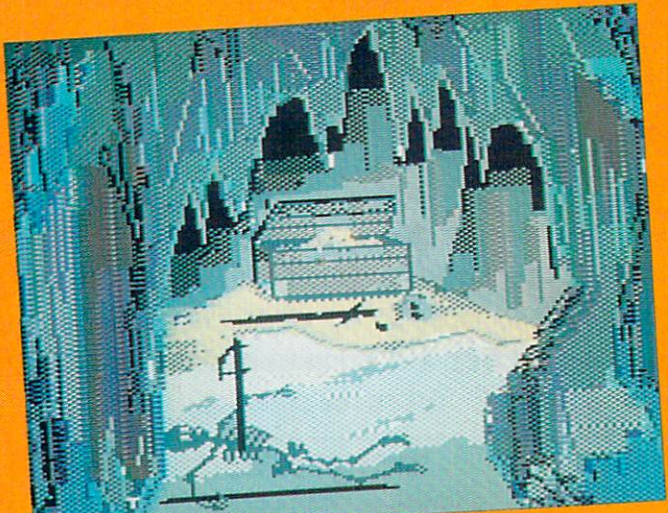
Art Gallery

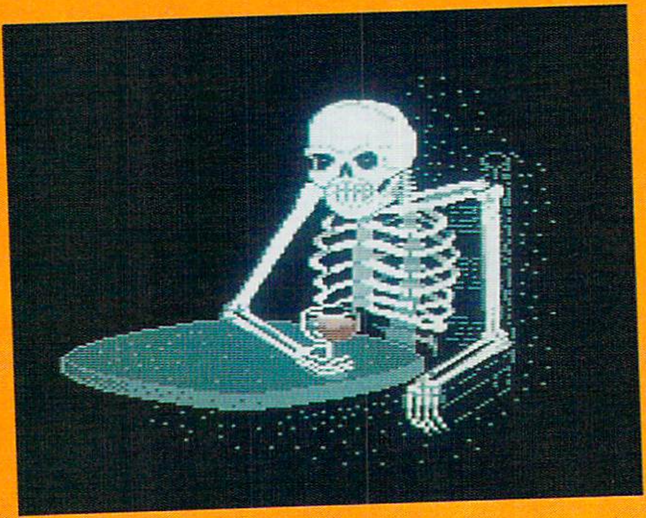


The *Art Gallery* offers the opportunity for fame and fortune to aspiring Commodore artists. Send your work on disk to *Art Gallery,* Ion International Inc., 45 West 34th Street–Suite 500, New York, NY 10001. Label each disk with the date of your submission, your name and address, the number of images on the disk, and the graphics or paint program used. Graphics produced on the Amiga are eligible for inclusion in *Ahoy!'s AmigaUser;* C-64, C-128, and Plus/4 images are eligible for inclusion in *Ahoy!* If your image is published, you will receive a free one-year subscription. Current subscribers will have their subscription extended by one year.

Note that the *Art Gallery* is not a contest. Published pictures are selected in an arbitrary and capricious fashion by the Art Director, based solely on their artistic merit.







Trick or treating may be a dying institution, but Halloween lives on. At any rate, you won't get cavities from these otherworldly images served up by Laura Palmeri, our bewitching Art Director. At top right is *Graveyard,* a favorite haunt of Cary Hildebrand (Morden, Manitoba). Didn't we see that tree in the Wizard of Oz? At middle right, a gruesome, toothsome *Jack-o'-Lantern* by Dale E. Clark (Flint, MI). (You have to admire pumpkins for their disposition. Would *you* smile after somebody carved your guts out to make pie?) At bottom right sits a bar patron who'll have a hard time holding his liquor: *Skeleton* by Daryl Maksymec (Regina, Saskatchewan). At bottom left is the second most frightening hole in the ground we've ever seen (the most frightening being the New York City subway)—*Cave* by Heinz Diekert (Vernon, BC). The first two images were created with *DOODLE!*, the latter two with *Koala.*

TIPS AHOY!

Continued from page 26

```
HEN C=C+1:GOTO 95
•70 IF A>128 AND A<134 THEN P=1:PRINT">";
:IF A=132 THEN Q=1
•75 IF N=21 AND Q THEN L%(F)=A:Q=0
•80 IF N=28 THEN L%(F)=A
•90 IF N=29 THEN L%(F)=L%(F)+256*A:N=0:P=
0:F=F+1
•95 IF C=2 AND A$=CHR$(128) THEN TB=1328
•100 IF A<32 THEN A$=CHR$(46)
•110 IF A>128 AND A<160 THEN A$=CHR$(58)
•120 PRINTA$;:GOTO 40
•130 PRINT:FOR R=0 TO F-1:PRINT L%(R),:T=
T+L%(R):NEXT:PRINT
•140 PRINTF" FILES",T" OCCUPIED BLOCKS",T
B-T" UNUSED BLOCKS[DOWN]":DIRECTORY" "
•150 PRINT"[DOWN]IF BLOCKS FREE <> UNUSED
 BLOCKS, VALIDATE DISK."
```

STARTING LINE

Those of you who own a Commodore 64 and a disk drive may have run into the familiar problem of forgetting the starting address of your machine language program. Most of the time, this starting address is also used to activate the program. *Starting Line* is the place to begin.

After RUNning the BASIC loader program below, you will be advised to the address used to activate the routine. You will know when *Starting Line* is active by the presence of an asterisk. Now any time you load a program from the disk drive, the starting address of that program will be displayed directly below the LOAD command.

Note: This machine language routine redirects the Kernal LOAD Routine Vector for its own purposes. If you plan on using a fast loading program, make sure that you activate *Starting Line* after you have activated the fast loader.

—Michael Jaecks
Alamogordo, NM

```
•10 REM ** STARTING LINE **
•20 S=53100:REM CHANGE TO RELOCATE ML
•30 FORI=0TO95:READA:POKES+I,A:B=B+A:NEXT
•40 IFB<>10742THENPRINT"ERROR IN DATA STA
TEMENTS!":END
•50 DEFFNHI(X)=INT(X/256):DEFFNLO(X)=X-FN
HI(X)*256
•60 FORI=0TO4:READA,B:POKES+A,FNLO(S+B):P
OKES+A+1,FNHI(S+B):NEXT
•70 POKES+18,FNLO(S+48):POKES+23,FNHI(S+4
8)
•80 PRINT:PRINT"SYS"S"TO ACTIVATE OR DEAC
TIVATE":PRINT"'*' = ACTIVE"
•90 DATA 173,95,19,208,28,173,48,3,141,94
,19,173,49,3,141,95,19,169,48,141,48,3
•100 DATA 169,19,141,49,3,169,42,32,210,2
55,96,141,49,3,173,94,19,141,48,3,169,0
•110 DATA 141,95,19,96,72,165,184,72,165,
185,72,169,5,168,166,186,32,186,255,32
•120 DATA 192,255,162,5,32,198,255,32,207
,255,170,32,207,255,32,205,189,169,5,32
•130 DATA 195,255,104,133,185,104,133,184
,104,76,0,0
•140 DATA 1,95,9,94,15,95,37,94,45,95
```

BASIC FLASHING SCROLL

You want to put some information on your screen for your latest game. You could print it with the PRINT statement, but that would be too boring. Why not put the information in a scroll routine? Better still, a flashing scroll routine. It would look good and you could display almost everything your 64K could hold. The only problem is that all the scroll routine programs that you can find are in machine language and all those data statements take years to type in. There must be an easier way.

And now there is! The *BASIC Flashing Scroll Routine*. This program can give you a scroll routine in no time. Here's how to use it. Type in the program as it is but substitute your message in the quotes in lines 30 to 50. If you want a bigger message, add in more lines between lines 51 and 59 in the same fashion as I have in lines 30, 40, and 50. Then run it. It couldn't be simpler. If one scroll routine is not enough to satisfy you, then erase the B$ from line 80 and you'll have all the scroll routines you can handle. Who said BASIC was slow?

—Nick Jankovic
Adelaide, Australia

```
•5 PRINT CHR$(14)
•10 POKE53280,14:POKE53281,14:PRINTCHR$(3
1)CHR$(147):H=53270:B$=CHR$(19)
•20 A$(.)=" [11"*"] [s T]IPS [s A]HOY! [1
9"*"] "
•30 A$(1)="[s B][s A][s S][s I][s C] [s F
]LASHING [s S]CROLL [s R]OUTINE. [s B]Y
[s N]ICK "
•40 A$(2)="[s J]ANKOVIC ([s K][s M][s T])
. [s R]EFER TO [s A]HOY! MAGAZINE"
•50 A$(3)=" [s O]CTOBER 1988 FOR INSTRUCT
IONS.[8" "]"
•60 FORA=.TO3:FORB=1TO40:FORC=7TO.STEP-1
•70 POKE53265,PEEK(53265)OR16:POKEH,C:NEX
T
•80 POKE53265,PEEK(53265)AND239:PRINTB$;M
ID$(A$(A),B,40-B);MID$(A$(A+1),1,B)
•90 GETE$:IFE$<>""THENPOKE53265PEEK(53265
)OR16:POKEH,8:PRINTCHR$(147):END
•100 NEXT:NEXT:POKE53265,PEEK(53265)OR16:
RUN
```

CBM MODULO

I have found a unique function called Modulo to be present on some other computers, but not the Commodore computers. It performs the integer division of two values and produces an integer remainder. Below is a representation of the MOD function for your Commodore.

By entering values for the variables A and B, the remainder is then displayed. One might ask of what use is the MOD function. It can be used for many mathematical purposes, but one unique use is finding which years are leap years. For example, by entering "1988,4" for the values A

and B, you find a remainder of zero. This indicates that 1988 is a leap year. You can go on to find leap years in the past or future by just looking for the remainder of zero. (*Note:* a value of four must be used for B when looking for leap years.)

—Michael Jaecks
Alamogordo, NM

```
·10 REM  THIS OPERATION PRODUCES THE
·20 REM INTEGER REMAINDER OF AN INTEGER
·30 REM DIVISION.
·40 PRINT:INPUT"A MOD B (ENTER A,B)";A,B:
B=ABS(B):C=INT(A/B):D=C*B:E=A-D
·50 PRINT:PRINTA;"MOD";B;"=";E
```

EXCLUSIVE-OR IN BASIC

There is no command available from BASIC 2.0 to compute the exclusive-or of two numbers. However, the command can easily be simulated with the following line:

```
1000 R=(N1 OR N2)-(N1 AND N2):RETURN
```

The variables N1 and N2 contain the variables to be EORed, and the result is stored in R. You might want to "play computer" and solve the equation for the four possible bit-pairs (0,0; 0,1; 1,0; and 1,1) to see how the equation works.

—Melvin Savage
Deer Park, MD

NO PROBLEM BASIC INPUT

The following routine allows you to get input in your programs without the usual problems associated with BASIC's input function. It is designed to work on the Commodore 128. It can be added to your program just like any other subroutine. When called, it prints a prompt and its own cursor. It will handle all those bothersome input problems for you. For instance, it does not allow the keys that typically mess up the input line such as CLR/HOME and the cursor keys to be entered. In addition, it allows for full use of the INST/DEL key to correct typing errors, but does not allow more to be deleted than have been typed. When the user hits the carriage return, the string variable CF$ will contain the string that was typed.

With this routine you can write your program without worrying about the user typing something that messes up the screen or puts the cursor in some weird place. I hope you find it useful.

—Paul Maioriello
Manalapan, NJ

```
·40000 PRINT "?";:PRINT"[c B]";:PRINT CHR
$(157);
·40010 CA$="":CF$=""
·40020 GETKEY CA$:IF CA$=CHR$(20) THEN 40
020
·40030 IF CA$=CHR$(17) OR CA$=CHR$(29) OR
 CA$=CHR$(145) OR CA$=CHR$(157) OR CA$=C
HR$(19) OR CA$=CHR$(147) THEN 40020
·40040 IF CA$=CHR$(13) THEN 40150
·40050 PRINT CA$;:CF$=CF$+CA$
·40060 PRINT"[c B]";:PRINT CHR$(157);
·40070 DO UNTIL CA$=CHR$(13)
·40080 :GETKEY CA$
·40090 :IF CA$=CHR$(17) OR CA$=CHR$(29) O
R CA$=CHR$(145) OR CA$=CHR$(157) OR CA$=
CHR$(19) OR CA$=CHR$(147) THEN 40080
·40100 IF CA$=CHR$(20) AND LEN(CF$)<1 THE
N 40080
·40110 PRINT CA$;
·40120 :IF CA$<>CHR$(13) THEN PRINT"[c B]
";:PRINT CHR$(157);
·40130 :IF CA$=CHR$(20) AND LEN(CF$)>0 TH
EN CF$=LEFT$(CF$,LEN(CF$)-1):ELSE CF$=CF
$+CA$
·40140 LOOP
·40150 REM (RETURN TO CALLING POINT)
```

COMPUTING A REMAINDER

Some math applications require you to find the remainder of a division, but once again, BASIC doesn't have a command for doing this. The easiest way I've come up with for computing a remainder is to use this function:

```
10 DEF FN MOD(X)=X-INT(X/D)*D
```

In the function, D is the divisor, and X is the dividend. The function can be used like this:

```
20 D=10:PRINT FN MOD(25)
```

The above line prints the remainder of 25/10.

—Melvin Savage
Dear Park, MD

Continued from page 22

tires and a winch to pull the truck out of mudholes are particularly wise investments for Georgia-bound gamers.

The joystick controls the player's 4 x 4 during the actual race. The gamer moves the joystick left and right to position the truck on the vertically scrolling course. Pulling back on the stick gradually slows the vehicle, while pushing it forward jolts it to an abrupt stop.

There are plenty of other trucks trying to cross the finish line, and they don't let the computerist's 4 x 4 cruise past them without a fight. The bumping and banging costs precious seconds and, if the player isn't careful, significant damage to the truck.

Electronic drivers who slam through other racing games at top speed are in for a shock. While it is important to maintain a fast pace, including making repairs and refueling as quickly as possible at the periodic checkpoints, sheer speed increases the amount of damage the trucks take from collisions and may cause the truck to run out of gas in the middle of nowhere.

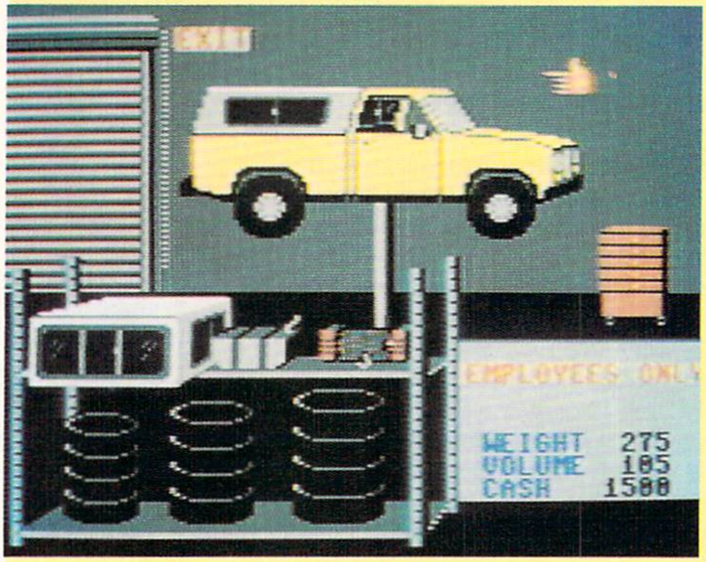
Paul Vernon's graphics, though simple, are very attractive. The truck's flips, rolls, and wheelies separate *4 x 4 Off-Road Racing* from more orthodox racing competitions.

Is this yet another recommendation of a computer racing game? It definitely is. *4 x 4 Off-Road Racing* is sure to bounce and jounce its way into the hearts of every computer driver.

Epyx, 600 Galveston Dr., P.O. Box 8020, Redwood City, CA 94063 (phone: 415-366-0606).

–Arnie Katz





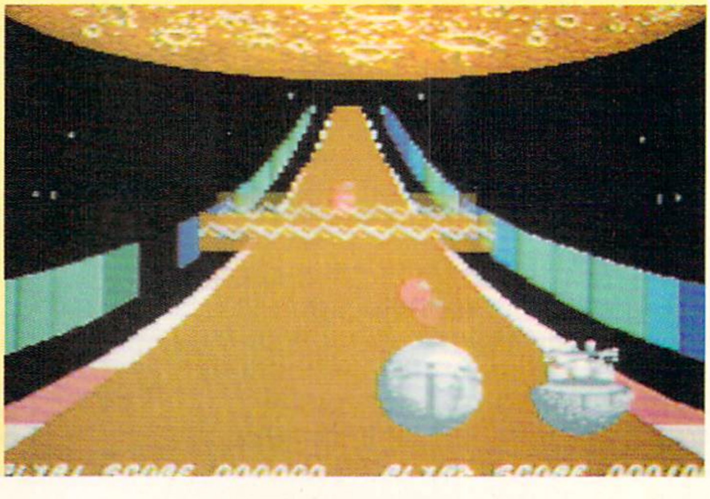
***4 x 4 (top and left) lets the driver customize his vehicle to meet the rigors of the chosen route. Supplies are purchased at the Auto Mart.* READER SERVICE NO. 109**

ROADWARS

Arcadia
Commodore 64
Disk; $29.99

This interesting and offbeat action contest is set in the far future when even moons can be linked together by cosmic highways under the control of master computers. The player manipulates a pair of "battlespheres," robotic globes equipped with laser cannon and shields, over this computerized outer space roadway.

The computer which controls this road had the ill grace to go rogue. It has started wiping out any vehicle which uses it. The battlespheres have



***The object of* Roadwars *is to keep the roads clear of obstacles. Battlespheres are used to destroy the panels that fire sparks.* READER SERVICE NO. 110**

been assigned the dangerous task of clearing the highway.

That's right, battlespheres. Even in solitaire versions, this is too big a job for a single droid, so the powers that be supply the player with a second sphere. If it's not under human control, the droid sphere "will act as a good but not a perfect partner," according to the instructions.

Players follow just behind the battlespheres as they road down this computerized gauntlet, using simple joystick commands to move the sphere left or right, and engage and disengage the shields.

The basic object of the game is to keep the road clear. On either side of the road are colored side panels which ordinarily serve to keep vehicles on the road. Now, however, under the malign guidance of the rogue computer, many of these panels have turned from their ordinary brown to a deep blue. The blue panels fire streams of deadly sparks across the road and must be obliterated by the battlespheres.

Additional obstacles appear as the game progresses, including laser-firing satellites and shield-destroying balls and chevrons.

The documentation is excellent and extensive, a refreshing change from most British software. In fact, the story background is, if anything, overdone. It goes on at length about the history of this planet, the lunar highways, the rogue computer, and the battlespheres. Then it blithely informs us: "Remember that in the bonus stages you are rewarded for killing your opponent [the second battlesphere], so do not cooperate too much." How about *that* for a game element out of left field? One must question the wisdom of imposing such a complex plot structure on a game, and then simply ignoring that same story halfway in.

The most serious flaw in *Roadwars,* however, is its visuals. The graphics are simply not up to current Commodore 64/128 standards. The color selection unsuccessfully contrasts the garishly colored road (orange, pink, and white with blue and brown side panels) with the washed out gray tone battlespheres. This tends to make the battlesphere, which *should* be the most visually prominent element onscreen, look insignificant.

The other graphic features, including explosions, sparks, and killer satellites, seem rather perfunctory. As a result, what should be a red hot roadway thriller comes off as a rather tepid action game.

Arcadia, 711 West 17th St., Mesa Business Center–Unit 9G, Costa Mesa, CA 92627 (phone: 714-631-1001).

–Bill Kunkel

FLOTSAM

Continued from page 31

This game recently received only 24 out of 100 points in a review by a British magazine, which also described the simulation as running so slowly that one got the impression the entire thing was written in BASIC. Contrary to your reviewer's comments, *Sub Battle Simulator* is not even in the same league as some of the other submarine simulations on the market, and I think there is no room for the "personal preference" argument. With many game prices at $30, I think you owe it to your readers to call a turkey a turkey, lest some unsuspecting reader actually go out and purchase a dismal product like this. Epyx has done some wonderful games in the past, but *Sub Battle Simulator* certainly is not one of them.

I find the rest of your magazine informative and entertaining, although it could use more reviews, and a little lightening up—after all we are talking about a $150 computer and a young user base. —Anthony Kulesa
Boston, MA

We've learned that the more strongly a reviewer praises or condemns a particular program, the stronger will be the rebuttal. And understandably so—emotions are not aroused by a middle-of-the-road opinion the way they are by a contradictory one. We've also learned that even when there seems to be "no room for the personal preference argument," we have to make room. But we'll also make room for the rebuttals—and we thank Mr. Kulesa for writing us with his.

To the staff of the BEST Commodore magazine:

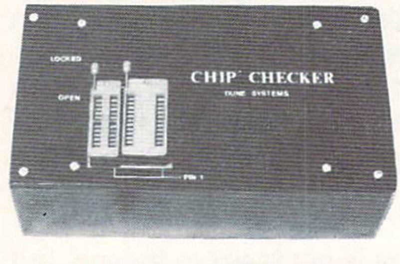
Let me start off by saying that I have never before been sufficiently motivated to write to any publication (computer-oriented or otherwise). But this time I felt it imperative to write and commend you on the job you have been doing. After letting my subscription lapse several months ago, my C-128 became more often used as a dust collector than a creative tool. I had become disenchanted with programming due to repeated (failed) attempts to move from BASIC to machine language programming. But then I once again picked up an issue (July '88)—and immediately had my spirits bolstered by the amazing feats of BASIC programming performed by the ever-more-prolific Cleveland M. Blakemore. Here were programs (in a MAGAZINE, no less!) of a quality and playability above that of commercial software! And in BASIC! (Okay, with a little ML support.)

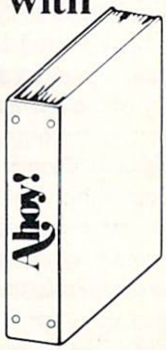
My first issue of this magazine was the July '86. I grew into programming with your mag and I cannot fully express my thanks in words.

Hopefully, with my confidence once again on the rise, I will finally work the bugs out of one of my games and get it in the mail to you.

Compared to the venerable *COMPUTE!* and its offspring, and the various other programming magazines, you produce a magazine of quality so far superior as to be in another league altogether. You have my complete loyalty. May my subscription never again expire. Once again, thanks.

—Jason Prince
Memphis, TN





REVIEWS

GEOS 2.0
Berkeley Softworks
Commodore 64
Price: $59.95

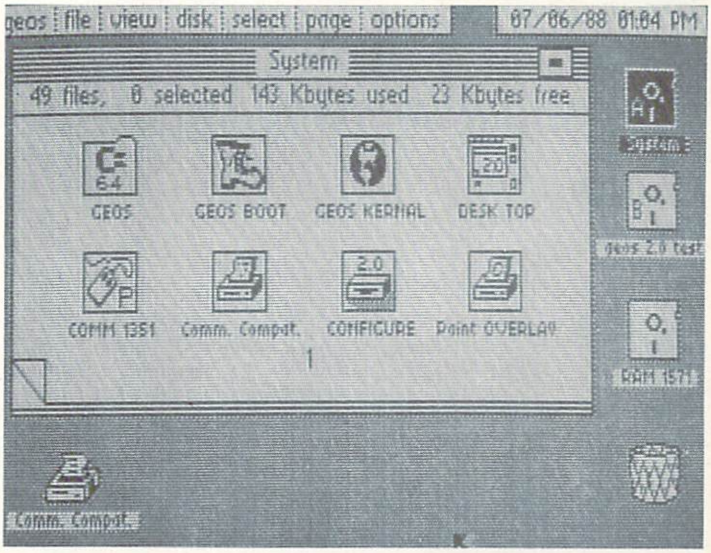
Has anyone out there not heard of *GEOS?* Has any one of you not had the opportunity to try out *GEOS?* On the off chance that some of our readers may have just bought their computer and have just looked at a Commodore specific magazine for the first time, we will briefly introduce *GEOS.*

The Graphic Environment Operating System *(GEOS)* is an easy to use disk-based desktop metaphor for the C-64 and C-128 computers. Since 1985, Commodore has included a copy of GEOS with every C-64 sold. For some reason, Commodore did not see fit to extend the same privilege to C-128 purchasers even after a version of GEOS was developed for the C-128. Thus, if you were a C-128 user and you wanted a copy of GEOS, you had to go buy it yourself.

As of this writing, Commodore intends to remove GEOS from the C-64 package and include it in the disk drive package itself. This makes some sense, as GEOS cannot be used without a disk drive. The program would only gather dust if for some reason you bought the computer without a disk drive. Of course, if you subsequently buy a second disk drive you will now end up with two copies of *GEOS* which should not cause any difficulties in any event.

The purpose of *GEOS* is to make the C-64 much easier to use. It replaces the computer's built-in, command driven BASIC and operating system with a graphic interface. Programs are selected by pointing and clicking at graphic representations, or icons, which are displayed on the screen as if they were spread about on a desktop. Instead of the keyboard, the primary input device becomes a joystick or mouse, or even a light pen or a Koala pad. *GEOS* has the built-in capability to manage the disk contents. Files can be copied and moved around, and entire disks can be duplicated by simply dragging the icons about the *deskTop.*

GEOS has become one of the most successful applications ever sold for the C-64. This success was virtually inevitable, as it was packaged with every computer sold for the past three years. Since its introduction, *GEOS* has gone through five revisions which were labeled 1.0 through 1.4. Each of these revisions fixed some bugs and added some features to the program. The last significant improvement was the inclusion of support for the Commodore 17XX series of RAM expansion modules for the C-64 and C-128. These modules let you add up to 512 kilobytes of external RAM to the computer. *GEOS* let you use this extra memory as a second disk drive to greatly speed up its operation.



A study of **GEOS 2.0's** *deceptively familiar screen reveals new menu choices, clock calendar display, and icons for three disk drives.* ***READER SERVICE NO. 120***

PHOTO: MORTON KEVELSON

The operation of *GEOS* is extremely disk intensive. Sections of program and data are continuously being swapped between the disk drive and the computer's RAM. Even though *GEOS* includes software which greatly speeds up the operation of the disk drive, disk speed is still the limiting factor in the use of the program. The 17XX series of RAM expansion modules help overcome this limitation, as they are much faster than the disk drive.

We have just received a Beta 4 release of version 2.0 of *GEOS.* The jump to a new version number signifies that *GEOS* 2.0 has some significant improvement over the 1.X series of the program. The remainder of this report will concentrate on the enhancements which have been made to merit this major upgrade.

Hardware Support

The most significant improvement in *GEOS 2.0* is the inclusion of support for the 1571 and 1581 disk drives. Although these drives have always been supported by *GEOS 128,* until now C-64 users have had to make do with the 1541 disk drive. As mentioned above, *GEOS* is extremely disk intensive. The use of the disk drive is essential for *GEOS* to do what it does in only 64 kilobytes of RAM. A typical GEOS application disk will include the copy of the desktop, a printer and input device driver, and some desk accessories such as the notepad or the photo manager in addition to the application itself and its data files.

Some of the newer *GEOS* applications, such as *geoPublish,* require a lot of space for both themselves and their data files. In fact the operation of *geoPublish* is very restricted if it has to run on an unexpanded C-64 and a single 1541 disk drive. Adding a second 1541 disk drive is beneficial but far from the ideal solution. In this case you have to split the application and its data files across two disks and some restrictions still remain.

The additional hardware support which is provided with *GEOS 2.0* should lift all operating restrictions for the foreseeable future. The 1571 dou-

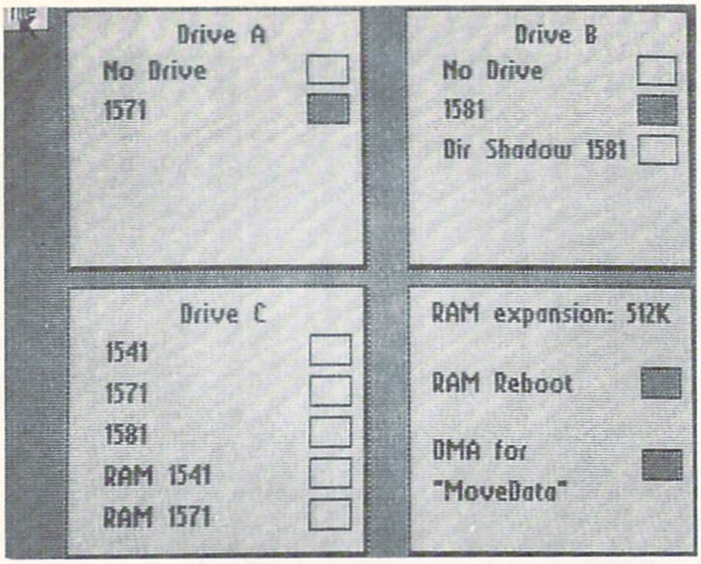


PHOTO: MORTON KEVELSON

The GEOS *Configure program's working screen provides a measure of insight into the disk drive and RAM options that are supported by version 2.0.*

ble sided disk drive has double the capacity of the single sided 1541 disk drive. This is enough to comfortably hold any *GEOS* application program and its data files. The 790 kilobyte capacity of the 1581 disk drive is nearly five times the capacity of the 1541. This is probably enough space to put all of your *GEOS* applications on a single floppy disk! On top of that, the 1581 is noticeably faster than a 1541 or 1571, even on the C-64.

One limitation in using the 1581 disk drive still remains. The *GEOS deskTop* has a capacity of 18 notepad pages which can each contain icons for eight files, or a total capacity of 144 files. This matches the 144 file capacity of the 1541 or the 1571 disk drives, but it is less than half of the 296 file capacity of the 1581 disk drive. We do not expect this to become an actual limitation for most users, as 144 files still represents an average file size of only 5½ kilobytes on a 1581 disk. Also, *GEOS 2.0* does not support the file partitioning and subdirectory capabilities of the 1581 disk drive. Perhaps both of these limitations will be overcome in a future release of *GEOS*.

If you only have a single 1581 disk drive, you will find *GEOS'* present whole disk copy facilities inadequate. The single drive DISK COPY program, on the *GEOS* system disk, did not work with the 1581 disk drive. We tried using the copy command on the *deskTop's* disk menu, but gave up after 15 disk swaps and nearly 20 minutes. As a result, depending on your hardware, you will either back up a 1581 disk a file at a time or use a stand-alone copy utility. Fortunately, the new multifile capability in *GEOS* makes it practical to back up several files at a time. What *GEOS* could use is a fast single drive backup utility which uses all available expansion RAM to minimize disk swaps.

Nearly as significant as the inclusion of support for the new higher capacity disk drives is the inclusion of support for a third disk drive on the *deskTop.* If you have them, the *deskTop* will now let you display up to three disk drive icons at a time. As in prior releases, one of these icons can still represent a RAM disk in the 17XX series expansion module. The limitation, in this case, is that only two of the drives can be active at any time. To activate the "C" drive, you have to click on its icon and drop it on either the "A" or the "B" drive. Incidentally, the icons for the "A" and "B" drives now display the letters "A" and "B" as well as the disk name. The icon for the "C" drive only shows the disk name.

When you drop the "C" drive (only its icon, of course) on the "A" or the "B" drive, it takes its place on the *deskTop*. The former "A" or "B" drive will then become the inactive "C" drive. Note that although the "C" drive is inactive, it remains logged in on the *deskTop.* If you intend to remove the disk from the drive you should close it before you swap it out to the "C" position. Activating the "C" drive is easier than it sounds. An even faster approach is to use oe of the new hot-key sequences such as the COMMODORE-SHIFT-A combination, which swaps the "C" drive with the current "A" drive.

Support of external expansion RAM has not been neglected. At this time, the limitations of a mere 512 kilobytes of external RAM become obvious. For example, it is possible to create two 1541 disk drives in RAM, but only one 1571 disk drive can be so emulated. Of course it is not possible to emulate a 1581 disk drive in only 512 kilobytes of RAM. A new version of the Configure program automatically shows the available options. For example, if you select a RAM 1541 you lose the ability to create a RAM 1571.

The new manual shows options for a shadowed 1571, but we were unable to bring this up in the Configure program. A shadowed drive keeps a copy of all that was read from it in RAM. This greatly speeds up disk access when a file or note pad page is accessed again. The Configure program also showed an option for a Dir Shadow 1581 which was not described in the manual. The Configure program also let us choose a 1541 or 1571 disk drive where an actual 1581 disk drive was hooked up. This caused the 1581's drive light to faintly flicker while the system locked up. Apparently the Configure program has not been finalized in this Beta 4 version of GEOS.

We also found a new 1351 mouse driver on the disk. This 1351 (a) driver is apparently an accelerating driver which responds to the speed as well as the position of the mouse. Fast mouse movements result in greater movement of the cursor than a slow mouse movement over the same distance. This provides for greater precision in cursor positioning. The old 1351 mouse driver, as well as drivers for the Koala pad and the Inkwell Systems light pen, are still included.

deskTop Improvements

Two changes are immediately apparent when you open up the version 2.0 *deskTop.* You may first notice the presence of a *deskTop* clock in the upper right hand corner, showing the date and time. Or, you may immediately spot the presence of seven menu items, at the upper left, where the old *deskTop* only showed five.

The new *deskTop* clock lets you set the date and time by clicking on the clock icon. After a brief disk access you can adjust the time and date by

simply typing in the changes over the existing settings.

Four of the menu selections, geos, file, view, and disk, remain the same as before. The special menu on the previous *GEOS* release has been replaced by select, page, and options menus. The contents of the geos menu remains essentially unchanged. The file menu contains two new choices: delete and undo delete. You will also notice that all of the choices in the file menu can be selected via a hot-key combination. The delete command lets you delete the selected files. It is no longer necessary to drag the file's icon to the waste basket. Also note our use of the plural in files. *GEOS* now supports multiple file selection. The undo delete command lets you retrieve the last file which was deleted, provided no other disk operations have taken place.

The view menu is the same as before. The disk menu shows some changes. As for the file menu, all of the disk menu commands now have a hot-key option for keyboard selection. The add drive command has been removed, as this function has been taken over by the Configure program. In its place you will find an erase command. This lets you delete all of the files on a disk without reformatting it.

We mentioned that *GEOS 2.0* supports multiple file selection. The select menu is part of this support. This menu adds three commands, all of which have hot-key alternatives. The all pages command selects all of the files on every page of the current disk note pad. The page files command selects all the file icons on the current notepad page. The border files command selects all the file icons on the border of the notepad. It is also possible to select groups of individual files by holding down the COMMODORE key while clicking on the files. Alternatively, files can be selected by pressing the COMMODORE key and one of the numbers 1 through 8. The numbers correspond to the eight icon positions on the note pad counting from left to right and from top to bottom. Multiple files can be selected by holding down the COMMODORE and SHIFT keys while pressing the number keys.

When several files are selected they can be all copied or deleted en masse. Clicking on any selected file invokes the multi-file ghost icon which represents the entire group. Just drag this icon around in the same way you would drag the individual icons. The multi-file feature cannot be used to copy groups of files to another disk with a single disk drive.

The page menu contains two new commands. The append command lets you add a disk note pad page after the current page. The delete command lets you delete the current note pad page and all files in it. If there are files on the page you will be able to cancel before the operation is carried out.

The options menu contains set clock, RESET, BASIC, and shortcuts. The RESET command reinitializes the *deskTop*. It lets you select a new disk,

READER SERVICE INDEX

Page	Company	Svc. No.
C-4	Access Software	266
46	Arcadia	110
49	Berkeley Softworks	120
7	Chip Level Designs	–
8	Commodore	273
16-19	Computer Direct	270
30	Computer Heroes	105
31	Computer Repeats, Inc.	115
10	Data East USA, Inc.	293
C-3	Data East USA, Inc.	114
48	Dune Systems	117
11	Electronic Arts	295
46	Epyx	109
48	ESP Corporation	116
12	First Row Publishing	104
12	Gamco Industries	288
8	Hayden Books	289
10	Honeywell	280
10	Honeywell	281
14	Howard W. Sams & Co.	282
12	Intracorp	287
12	Konami	103
12	Lucasfilm Games	102
4,5	Lyco Computer	268
10	Mediagenic	294
56	Microillusions	121
11	Mindscape	297
13	Mindscape	278
13	Mindscape	279
20	Mindscape	108
43	Montgomery Grant	269
C-2	Origin Systems	240
13	Pageant Publishing Co.	277
11	P.A.V.Y. Software	111
13	Phoenix Electronics, Inc.	112
10	Prentice Hall	291
55	Q-Link	106
11	Rainbird	298
12	Readup Inc.	286
30	Remsoft	272
14	Rent-A-Disc	118
12	Scorpion	101
8	Simon & Schuster	276
11	Simon & Schuster	300
11	Sir-Tech Software	299
21	Software Simulations	161
24,25	Software Support Int'l.	265
10	TAB Books Inc.	290
14	TAB Books Inc.	283
9	Tevex	271
8	The Hunter Group, Inc.	274
11	Three-Sixty Pacific, Inc.	296
20	Three-Sixty Pacific, Inc.	107
14	Verbatim News Services	284
45	Wedgwood Rental	119
8	Wico Corporation	275
39	Xetec, Inc.	113
6	*Ahoy!* Disk Magazine	–
34	*Ahoy!* Subscription	–
41	*Ahoy* Disk	–
47	*Ahoy!* Access Club #2	–
48	*Ahoy!* Binders	–
53	*Ahoy!* Access Club #1	–
57	*Ahoy!* Back Issues	–

The publisher cannot assume responsibility for errors in the above listing.

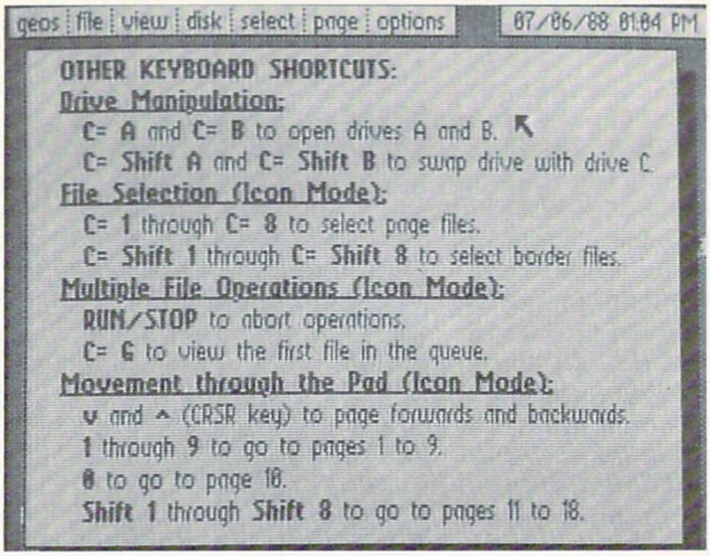


PHOTO: MORTON KEVELSON

A new GEOS help screen shows the hot-key options which are not available in the menus. Version 2.0 allows users to do almost everything via hot-key commands.

such as for copying to, without having to close the presently open disk. The BASIC command exits *GEOS* and returns control of the system to the built-in BASIC. The shortcuts command, which is not listed in the manual, displays a screen full of additional hot-key commands which are not shown on any of the other menus. *GEOS* version *2.0* lets you do nearly everything with hot-key commands. As you become familiar with the system, your reliance on the input device should decrease. In general we found the hot-key commands to be faster than performing the same operations with the mouse.

Conspicuous by its absence from any menu was the command for booting Q-Link directly from *GEOS*. Nevertheless, Q-Link still remains the official telecommunications network for GEOS users.

Applications

Along with the *GEOS 2.0* you will get new applications software. In particular, the accompanying word processor has been upgraded to *geoWrite 2.1,* the same one which is supplied with the *geoWrite Workshop.* In fact, with exception of the *geoFont* application, *GEOS* now includes the entire contents of the *geoWrite Workshop* and *geoSpell.* These packages have been sold separately for use with *GEOS*. We do not have the space to present a complete review of these applications; however, we will highlight some of their features.

GEOS is a graphics-oriented system. Documents which are created by the system are treated as large bit maps for printing on a dot matrix printer. *GeoWrite 2.1* lets you use the full 8" width of the printer for its documents. In comparison, the previous release of *geoWrite* maintained 1" margins on each page. When you tell *geoWrite 2.1* to use the full page width, it will automatically reformat the current document. The new format is not compatible with earlier versions of *geoWrite*. Keep this in mind if you intend to pass a copy of the document to someone who does not have the new version of *geoWrite.* Other notable improvements in *geoWrite 2.1* include headers and footers, the ability to format individual paragraphs, and search and replace.

The other applications provided with *GEOS 2.0* include *geoSpell,* a full-featured spelling checker, and *geoMerge,* a mail merge program. The Text Grabber utility lets you convert text files from other word processors to *geoWrite* format. The supported file types are generic PETSCII, *Easy Script, Paper Clip, SpeedScript,* and *Word Writer.* The Paint Drivers application lets you convert a page from a *geoWrite* document into a *geoPaint* file format. This lets you illuminate your text with *geoPaint,* the graphics application which is available separately. If you happen to have access to an Apple LaserWriter printer, you will be able to print your *GEOS* text and graphics documents on it with the geoLaser application program.

We found one new Desk Accessory on the *GEOS* disks, the Pad Color Manager. *GEOS* now lets you code file icons as well as the background and foreground of the disk note pad on the *deskTop*. The color combinations are based on file categories such as data files, desk accessories, and System files. The Pad Color Manager provides 16 categories of *GEOS* files for which you can set any of the 16 available colors. Only the files' icons are colored. The file names remain in the *deskTop* pad foreground color. Three default color maps are provided, or you can save your own settings. The settings take effect when the disk is logged in. Although color is not essential to the operation of the *deskTop*, it adds a nice touch. The Pad Color Manager lets you adjust the *deskTop* colors to suit your tastes and the characteristics of your video display.

The *GEOS* manual now numbers over 300 pages, more than double the size of the previous edition. It is supplied as a bound trade size paperback with holes punched for a three ring binder. We miss the spiral binding, which was used for the last edition, which let the manual lay flat on the table. Since the book is nearly an inch thick and firmly bound, we were not sure what the holes were for. Perhaps we are supposed to tear out the pages for inclusion in a loose leaf binder.

Part of the manual's bulk is due to the inclusion of the manuals which accompanied the *geoWrite Workshop* and *geoSpell* packages. These manuals were not simply copied over. They were all edited and updated prior to being included in the new *GEOS* manual. In fact, the entire manual reads better than ever before. Still included are introductory and tutorial chapters, along with separate chapters for each individual application.

Conclusion

GEOS 2.0 represents a significant improvement over the previous release. Although major emphasis has been on the use of additional hardware, the user interface has not been neglected. The addition of numerous hot-key combinations for most *GEOS* operations will make GEOS more comfortable for experienced users. The inclusion of *geoWrite 2.1* and *geoSpell* makes *GEOS* a fully functional word processing package right out of the box.

We still feel that *GEOS* requires the right hardware to really fly. Although you can get by with one disk drive, our

GET ACCESS TO REAL BUYING POWER... WITH THE Ahoy! ACCESS CLUB!

Subscribing to *Ahoy!* and/or *Ahoy!'s AmigaUser* has always made sense—for you and for us. We get to keep more of your money when we cut out the middlemen (our distributor and your newsdealer), and we kick some of the savings back to you with a discount rate (see page 34).

And now you can save even more—in fact, you can save the cost of your subscription many times over!

The *Ahoy!* Access Club, launched in January 1986, has been expanded to offer its members even more clout in the Commodore marketplace. And for a limited time, membership will be awarded free to subscribers!

Here are some of the ways the *Ahoy!* Access Club can boost your buying power:

- The *Ahoy!* Access Club *Clipper*, published 12 times a year, contains exclusive discount offers on products advertised in *Ahoy!* and *Ahoy!'s AmigaUser*. Participating vendors offer reductions of 10%, 20%, 30%, or more on selected items, free bonus merchandise, and special close-out and combo offers not advertised elsewhere—all for Club members only!
- The *Ahoy!* Access Club Card is your ticket to discounts at participating computer software, hardware, and book dealers, and reduced admission at Commodore and Amiga conventions and swap meets across North America. (Details are found in each issue of the *Clipper*.)
- The *Ahoy!* Access Club BBS (modem required) offers continuously updated information on new offers available through the Club, as well as late-breaking industry news, corrections and updates to articles in *Ahoy!* and *Ahoy!'s AmigaUser*, and free electronic mail facilities. Operation is 24 hours a day, 7 days a week—for Club members only!

If you are presently a subscriber to *Ahoy!* or *Ahoy!'s AmigaUser*, your membership has already been activated. You'll receive the *Ahoy!* Access Club *Clipper* bound into each issue.

If you're not a subscriber, fill out and return the postpaid card bound between pages 50 and 51 today!

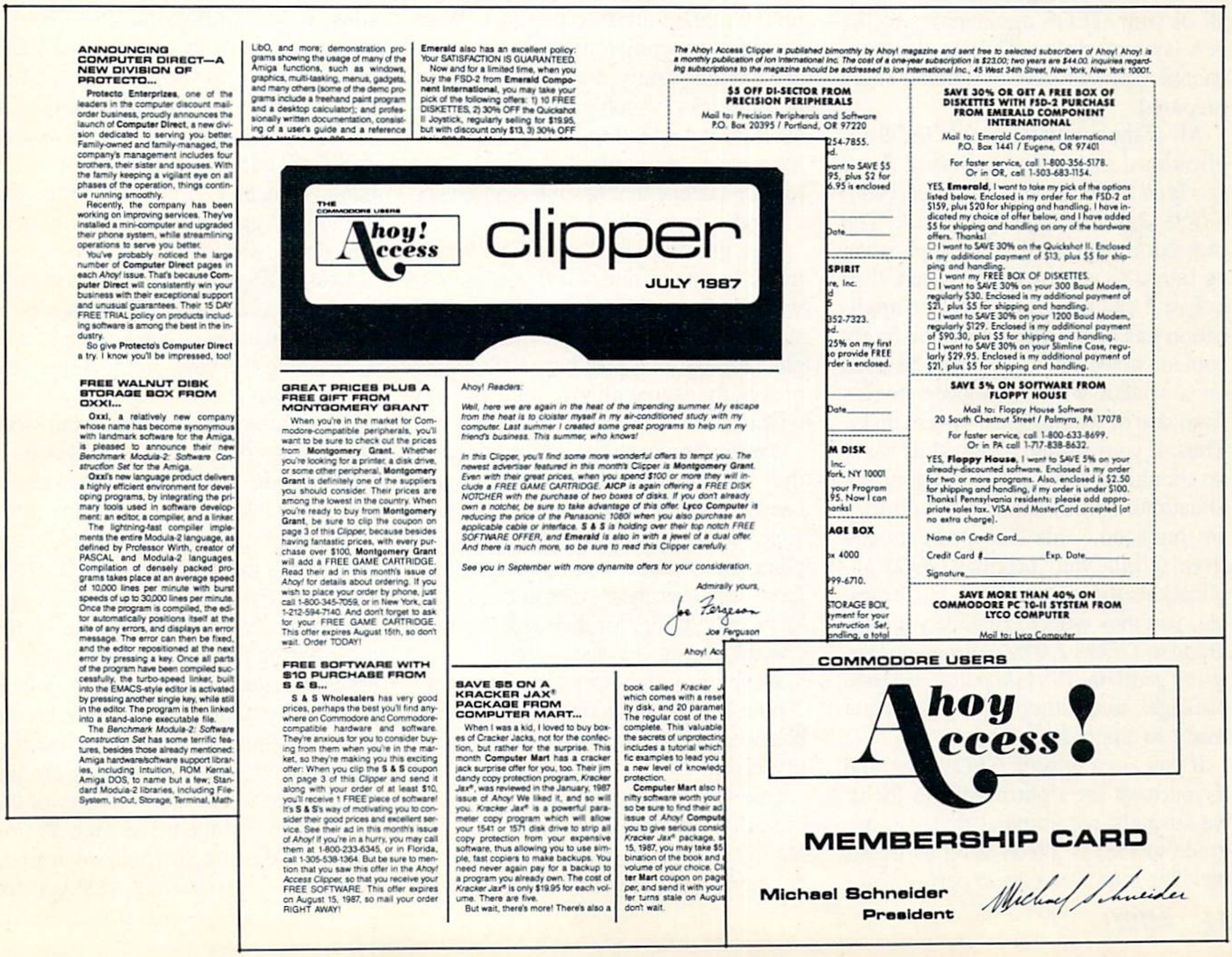
Let everyone else pay list price. You don't have to...when you have Access!

LOCK IN YOUR FREE MEMBERSHIP NOW!

Ahoy! Access Club membership will be included free with your paid subscription for a limited time only.

You are guaranteed free membership for the duration of any subscription paid for prior to December 31, 1988. After that date, a small additional charge may be levied.

So why not lock in free membership for as many years as you wish by extending your subscription now?



recommendations are that the minimum system should include two disk drives and a 1351 mouse. You will need a 1541 or a 1571, preferably the latter, as *GEOS* is supplied on three 5¼" disks. For a second disk drive we urge the use of the 1581. Its large capacity and extra speed make it ideal for use with *GEOS*. We also recommend the inclusion of one of the Commodore 17XX series modules. For C-64 users, the 1764 is a good choice; however, if you can come up with a replacement power supply you should consider the 512K 1750 RAM expander. We were running our copy of *GEOS 2.0* on a C-128 with two 1571 disk drives, a 1581 disk drive, and a 512K RAM expander. *GEOS* never had it so good.

GEOS still uses copy protection on its system disks. Although an original and a backup disk are provided, they cannot be duplicated. On the plus side, you only need one of the original disks when you boot the system. After booting, all work can be done from a copy of the system disk. The first time you use the system disk it installs itself and the backup copy on your system. You are also given the opportunity to link all of your GEOS applications to the new system disks. This is the only chance you will have to do this, so be prepared.

All of the separately sold *GEOS* applications, including *geoWrite 2.1* and *geoSpell* which are supplied with *GEOS 2.0*, are copy protected. The protection requires that the application be linked to your original system disk before it can be used. Once the application has been linked it can be freely copied; however, it can only be used on a system which has been booted from one of your original system disks. Thus, if your system disk and its backup should be damaged, all of your applications will be unusable until they are replaced. This also prevents you from selling your original *GEOS* applications to another user. For example, you may wish to finance your upgrade to *GEOS 2.0* by selling your original *geoWrite Workshop* and *geoSpell* packages to another user who is not ready to upgrade.

If you are a serious *GEOS* user, and if you have the right hardware for it, we strongly recommend that you upgrade to *GEOS 2.0* as soon as possible. The improvements in performance are well worth the cost. If you are new to the C-64 and are looking for an easy to use, intuitive graphic operating system, rest assured that *GEOS 2.0* is the best *GEOS* yet.

Berkeley Softworks, 2150 Shattuck Avenue, Berkeley, CA 94704 (phone: 415-644-0883). *—Morton Kevelson*

TYPING TEACHER
Future Age Computers
Commodore 64
Disk; $29.95

There has probably never been a better typing teacher than the personal computer. You practice when it is convenient; you progress from one lesson to another at your own pace; speed drills can be taken whenever the urge moves you; and no one but you need see your embarrassing early scores.

Moreover, computerized typing teachers use a variety of displays to aid your drills and lessons, and because of this they do a good job of keeping you from peeking at the keyboard to see what your fingers are doing.

That personal computers excel at teaching typing has been known for some time, and the variety of programs on the market attest to that fact. With so many programs available, choices are too frequently made on the basis of gimmicks: which program will better hold my attention, the one that allows me to type missiles at space aliens or the one that has me typing cars around a race track?

The gimmick to *Typing Teacher* is that it has no gimmicks. If you've ever wished you could walk into a software store and buy a straightforward, simple, bare-bones typing tutor without alphabet-menacing aliens, you can stop wishing.

Typing Teacher is so uncomplicated that the instructions for use occupy barely more than a single page, the balance of the manual being used to explain punctuation and spacing rules. Don't skip over these rules thinking you know everything, for they will become crucial during the speed drills.

In spite of its simplicity, however, *Typing Teacher* has a bit of trouble with balance that almost amounts to an identity crisis.

One of the first questions to be asked of such a program is whether it is going to teach you to type on a typewriter or on a computer. If you think there isn't enough of a difference to matter, take another look at your computer keyboard. Certainly the letters are the same, and they are probably arranged in the common QWERTY pattern. Yet computers have some characters not found on common typewriter keyboards, and while some of these (such as "<" and ">") may only be of interest to programmers, they are still there and need to be addressed. Fortunately, *Typing Teacher* does recognize most of these extra keys.

The second question is whether the tutorials and speed drills will function as typewriters or word processors, and it is in this area that *Typing Teacher* decides it is a semi-typewriter: words continue from one line to the next, but do not wrap around. As an example, only the first six letters of "respect" might fit on the line, leaving the final "t" to appear on the next line. It may not seem important, but when you're going through a speed drill and are trying to type exactly what you see on the screen, such breaks will drive you crazy. More about that later.

You begin at the beginning, although it is possible to select any of 16 lessons. It is also possible to select instruction and drills for either QWERTY or Dvorak keyboards, though few Commodore owners will have use for the latter. The first lesson drills you on A,S,D,F, and G—a drill for the left hand. At center screen, the letter you are to type appears and, as quickly as you do it, the next letter appears.

Lesson Two adds keys for the right hand, and after that you'll begin exploring combinations of hand usage as well as learning the keys on the upper and lower rows.

As each lesson is finished, your word count and error count is shown on a final screen, and then you may either repeat the lesson or go on to the next.

In speed drills, a partial paragraph is shown in the upper half of the screen and your own typed responses are shown in the lower half. One of the most boring things about typing teachers is the quality of prose with which you must practice: "The quick brown fox...." is interesting the first time it is encountered, but seldom again. In using for drill work a short history of the development of the typewriter, *Typing Teacher* did manage to hold my interest.

You must type exactly what you see



"A direct line, that is, from your home to Commodore® Headquarters via Q-Link, the telecommunications service that's everything a Commodore owner could hope for.

Using a Commodore 64® or 128™, disk drive, modem, and the Q-Link software, you're connected to inside information and help from the programmers and designers here at Commodore who built your machine. It's the best way I know to get you the answers quickly and personally.

Q-Link is also your link to leading Commodore software publishers and their wares, to over 15,000 public domain software programs you can download and keep, to teachers who'll help your kids with their homework, and to clubs, contests, games, and a host of other services that run the gamut of your imagination.

Experience it for yourself. And see why I've put my company on the line for you."

Get a FREE Commodore modem and FREE Q-Link software when you join Q-Link.

Call toll-free 1-800-782-2278 ext. 1560

or mail this coupon to **Q-Link, 8619 Westwood Center Drive • Vienna, VA 22180**

Choose one:

☐ **I need a modem.** Send me the FREE Commodore 300 baud auto-dial modem and the FREE Q-Link software. Charge me now for my first 4 months of Q-Link membership at $39.80 ($9.95 per month).

☐ **I already have a modem.** Send me the FREE Q-Link software and charge me now for my first month of Q-Link membership at $9.95.

Full Name ____________________

Street Address ____________________

City ____________________ State ________

Zip ____________ Phone (______) ____________

Signature ____________________

Choose your method of payment:

☐ Please charge my credit card. ☐ MasterCard ☐ Visa

Acct. # ____________________ Exp. ________

☐ Check enclosed.

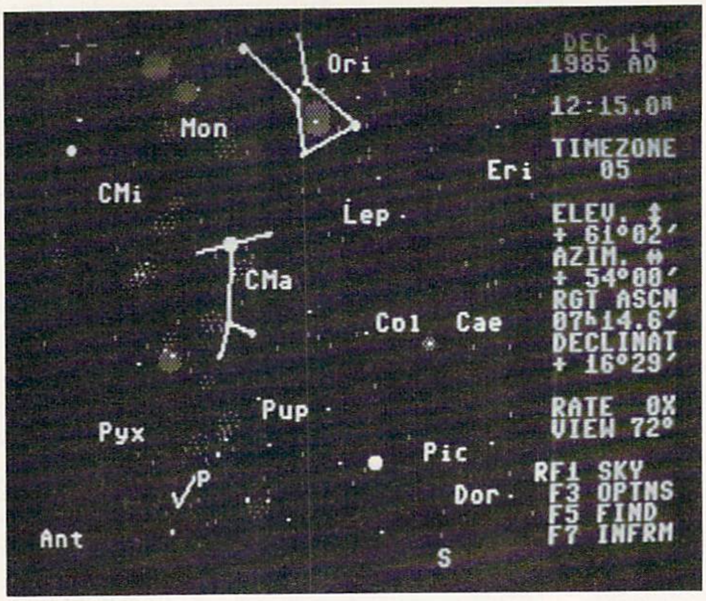
30 DAY MONEY-BACK GUARANTEE-YOU RISK NOTHING. Try Q-Link for 30 days. If you're not completely satisfied, send us (within 30 days of registering online) your letter of cancellation along with the Q-Link software and modem you received, and you'll get a full refund of your Q-Link monthly membership fee.

8619 Westwood Center Drive
Vienna, VA 22180
1-800-782-2278 ext. 1560

This offer is only valid for new members who respond to this advertisment.
Offer expires March 31, 1989.

Q-link SM

The Commodore Connection.



Sky Travel *lets you view the stars from any point in the world, and at any point in time up to 10,000 years in the past or future.* *READER SERVICE NO. 121*

on the screen, down to the last space. Paragraphs always begin at the left margin, without indentation; the words break at the right without wrapping around; and if you forget to space after a colon (or forget any other punctuation convention outlined in the manual) you're in trouble for the rest of the paragraph. It seems rather unfair that one error in the middle of your text will cause all the rest of the text to be in error, but it did happen to me more than once.

Since the emulation here is of a semi-typewriter, you use the RETURN key only when you're ready for your word count. Press it and you'll quickly be given your words-per-minute rate, number of errors, corrected score, and so on. These results can be saved on a separate data disk, for later comparisons of improved speed and accuracy.

While documentation for *Typing Teacher* is slight, there is no need for it to be any more: there are enough on-screen instructions to guide you through all the lessons. In reading the manual, you'll find that *Typing Teacher* claims to have a response fast enough to support typing speeds of up to 350 words per minute. And yet it says nothing about caring for fried fingers.

There are a lot of good, simple points to *Typing Teacher.* I think I'd like it better if, in speed drills, it behaved less like a semi-typewriter, and more like a computer/word processor.

In sum, however, it may be that *Typing Teacher's* claim to fame will lie in the fact that it can teach typing without frills, fancy stuff, and aliens.

Future Age Computers, 3 McCracy Drive S.W., Rome, GA 30161 (phone: 404-235-4948). *—Ervin Bobo*

SKY TRAVEL
Microillusions
Commodore 64
Disk; $49.95

For a long time I've been dismayed about the principles of software marketing: the "here today, gone tomorrow" philosophy that gives very few programs a shelf life of more than six months. Those of us who have owned computers for a while and who make regular trips to the software store find the practice adequate, but what of the new users who may never experience the fun of *Gorf* or the challenge of *Jupiter Landing?*

Fortunately for those of us who agonize over such things, the past year has seen many programs reissued in economy packs, sometimes with a name change. But missing from these collections are programs originally published by Commodore. And in spite of what you may have thought from the atrocious packaging, Commodore did publish some winners.

Among these was *Sky Travel,* which I first saw in 1985. A new edition has just been published by Microillusions. Granted that it is a repackaging of an old program—it is a program that was never as successful as some of us thought it should have been. Now that Microillusions has become an affiliate of Activision, it may be that *Sky Travel* will be better distributed than when published by Commodore.

Billed as an "all-encompassing astronomy program," *Sky Travel* more than lives up to its name by providing you with the computer equivalents of a powerful telescope, an SST airliner, and a time machine.

Select the Map mode and you can set the cursor at any point in the world from which you may wish to view the sky. And if you know the latitude and longitude, you can set a finely tuned location. (If you don't know the coordinates, an appendix lists those for most of the world's major cities.)

In selecting time, *Sky Travel* allows you to set month, day, year, and hours and minutes—and displays surprising accuracy for any date up to 10,000 years in the past or 10,000 years in the future.

As an example of those two powerful features, I'll tell a story I've told before: On the evening I purchased *Sky Travel*, our family talk on the way home was of stars, and naturally someone looked at the sky and wondered aloud about a bright object near the moon. When dinner was finished, I booted *Sky Travel* and set the proper coordinates and the current date, and chose a time about 90 minutes earlier.

It happens that the default view of the program is the default view of one standing in my driveway, and the object was there. Using the Inform feature, I found that the bright light was Mars.

Well, I immediately rushed the family back outdoors for another look, ready to impress them with my new knowledge, but after telling them what they were looking at I grew silent and thoughtful.

The first and best book of science fiction I ever read was Ray Bradbury's "The Martian Chronicles," but in spite of that and all the books that followed, I was an astronomical idiot. This may not have been the first time I'd seen Mars, but it was my first time to see it and know what it was.

Thanks to *Sky Travel.*

Once you've set time and location, you return to the Sky mode. Your monitor displays a small section of the sky with symbols and names for constellations and planets filled in and with lines connecting the constellations. Any or all of these embellishments can be de-

Continued on page 74

BACK ISSUES OF Ahoy!

#2—FEB. '84 Illustrated tour of the 1541! Artificial intelligence! Synapse's Ihor Wolosenko interviewed! String functions! And ready to enter: Music Maker Part II! Screen Manipulation! Night Attack! Relative Files!

#3—MAR. '84 Anatomy of the 64! Printer interfacing for VIC & 64! Educational software series begins! VIC game buyer's guide! And ready to enter: Address Book! Space Lanes! Random Files on the 64! Dynamic Power!

#5—MAY '84 Future of Commodore! Inside BASIC storage! Memory management on the VIC & 64! Guide to spreadsheets! Hurray for arrays! And ready to enter: Math Master! Air Assault! Biorhythms! VIC Calculator!

#7—JULY '84 MSD dual disk drive! Database buyer's guide! Training your cursor! Screen displays! File Sleuth! Users Groups! And ready to enter: Renumbering! Checklist! Math Defender! Brisk! What's My Job?

#8—AUG. '84 Choosing a word processor! Computational wizardry! Creating your own word games! Sound on the 64! And ready to enter: Micro-Minder! Directory Assistance! The Terrible Twins! Words Worth!

#9—SEPT. '84 Program your own text adventure! Build a C-64 cassette interface! Video RAM! Word processors, part II! And ready to enter: Salvage Diver! DOS! Sound Explorer! The Castle of Darkness! Base Conversions!

#10—OCT. '84 C-64 graphics programs! Bit-mapped graphics! Joystick programming! Graphics processing! And ready to enter: VIC 40 Column Operating System! BAM Read & Print! Emerald Elephant! Lawn Job!

#11—NOV. '84 Music programs & keyboards for the 64! Graphics feature continues! 2-D arrays! And ready to enter: PTE word processor! Block Editor! Alternate Character Set for the 64! Tunnel of Tomachon!

#12—DEC. '84 Buyer's guide to printers! 1525 printer tutorial! Fast graphics with custom characters! User Guide to KMMM Pascal! Diving into BASIC! And ready to enter: Construction Co.! Space Patrol! Cross Ref! **SOLD OUT**

#13—JAN. '85 VIC and 64 OS exposed! Sprites! Insert a 1541 device disconnect switch! Ghostbusters! And ready to enter: Ultra Mail! Music Tutor! Alice in Adventureland! Midprint! To the Top! Tape/Disk Transfer! **SOLD OUT**

#14—FEB. '85 Printer interfacing! Multicolor sprites! Modems! Bulletin Boards! Theory of game design! Buying a modem! And ready to enter: Futurewar! Fontasia! VIC Eraser! Insurance Agent! Flankspeed! Telelink 64!

#15—MAR. '85 Creating multiscreen gameboards! Inside the Plus/4! Commodore DOS! And ready to enter: Old Routine! Programmable Functions! Automatic Line Nos.! Home Budget! Salmon Run! Numerology!

#16—APR. '85 Assembly language column begins! Programming the joystick! 1541 disk drive alternatives! The Kernal! And ready to enter: Hop Around! Faster 64! Booter! Elecheck! BASIC Trace! Space Hunt!

#17—MAY '85 Disk drive enhancements! Install a reset switch! Assembler escapades! And ready to enter: Super Duper! Two-Column Directory! DSKDU! Raid! DOS Plus! Font Editor! Tile Time! Interrupt Wedge!

#18—JUNE '85 Music & graphics entry system! How modems work! Inside the 6510! And ready to enter: Quad-Print! Mapping 4.4! Towers of Hanoi! Speedy! Duck Shoot! Bit Dumping! Screen Magic! 6510 Simulator!

#19—JULY '85 PROM Programming! 3-part harmonies on VIC/64! Speeding pixels! And ready to enter: Auto-Append! Script Analysis! Wizard of Im! Lucky Lottery! Brainframe! Etch! Printat! Autos: Leasing v. Buying!

#20—AUG. '85 Inside the 128! Real-world simulations! Sound effects! Modems! And ready to enter: Windows! Formatter! Sound-a-Rama! Screen Dump! Selectachrome! Disintegrator! Fidgits! Gators N Snakes!

#21—SEP. '85 Inside the 1571 drive and 128 keyboard! Sprite programming! And ready to enter: Fastnew! Go-lister! File Lock! Dragon Type! Superhero! Auto-Gen! Moxey's Porch! Fish Math! Ahoy!Dock! Invective!

#22—OCT. '85 Create cartoon characters! Infinitesimal intrigue! Secrets of copy protection! And ready to enter: Shotgun! Maestro! Solitaire! Mystery at Mycroft Mews! Gravinauts! 1541 Cleaning Utility! Shadey Dump!

#23—NOV. '85 Guide to adventure gaming! ML sprite manipulation! BASIC for beginners! And ready to enter: Lightning Loader! Knight's Tour! Chopper Flight! Rhythmic Bits! Instant Bug Repellent! File Scout! Slither!

#24—DEC. '85 Speech synthesizer! The IBM Connection! The year's 25 best entertainments! And ready to enter: Gypsy Starship! Directory Manipulator! Cloak! Gameloader! Jewel Quest! Lineout! Santa's Busy Day!

#25—JAN. '86 Build a speech synthesizer! Survey of sports games! And ready to enter: Martian Monsters! Streamer Font! Microsim! Haunted Castle! Knockout! Infraraid! Alarm Clock! Memory Check! Scratch Pad!

#26—FEB. '86 Windows! Build an autoexec cartridge! Align your 1541! Survey of flight simulators! Structured programming! And ready to enter: Arena! Head to Head! Crabfight! Treasure Wheel! Character Dump!

#27—MAR. '86 Programming educational games! Memory dumpers! Choosing a copy program! Custom characters! And ready to enter: Ahoy!Term 128! Trivia Game Maker! Brickbusters! Easy Lister! Programmer's Aid!

#28—APR. '86 Comet catching! Survey of action and strategy games! Screen dumping! And ready to enter: Chrono-Wedge! Mr. Mysto! Air Rescue! Notemaker! Screen Window! JCALC! Hidden Cavern! Swoop!

#29—MAY '86 128 graphic bit map! Epyx strategy guide! 128 commands! ML music programming! And ready to enter: Bigprint! Star Search! Failsafe! English Darts! Ski Folly! Free RAM Check! Alchemist's Apprentice!

#30—JUNE '86 Debugging dilemmas! Public domain software! Winning at Ultima! Computer Aided Design! And ready to enter: LazyBASIC! Got A Match? Star Strike! Queen's and Bishop's Tours! Shaker! Trackdown!

#31—JULY '86 Inside the Amiga! Conditional branching! Chess programs! 128 and 64 DOS! And ready to enter: Screen Sleuth! Skull Castle! Head-on! Nebergall Run! Wordcount! Crazy Joe! Fidgits' Music School!

#32—AUG. '86 Inside the Amiga, part II! Approaching infinity! C-64 war simulations! Pascal for beginners! ML graphics! And ready to enter: Reversi! Highlight! Disk Cataloger! Meteor Run! Trim! Step On It! Flap!

#33—SEPT. '86 Windows and viewports! Sound & music on the 64! COMAL! And ready to enter: The Last Ninja! Speech64! Multi RAM! Dogcatcher! Trapped! Matchblocks! Variable Manager! Dual Dump! Mine Canyon!

#34—OCT. '86 Build a digital oscilloscope! ML speed techniques! And ready to enter: Vault of Terror! Quick Change! Penguins! Attack Force! Disk Checkup! Dvorak Keyboard! Mountaineer Mack! 128 to 64 Autoboot!

#35—NOV. '86 C-128 shadow registers! Data file handling! PROMAL! Habitat! And ready to enter: Teleporter! 128 RAM Check! Discs of Daedalus! Guardian! Tenpins! Syntax Patrol! Deluxe List! Long Lines! Detonation!

#36—DEC. '86 File manipulation! C-128 shadow registers! Football games! And ready to enter: The Artist! Minotaur Maze! Mouse in the House! Lazy Source Code! Rebels and Lords! Speedway! The Editor! Micro City!

#37—JAN. '87 Pointers and the monitor! Best games of '86! DOS for beginners! And ready to enter: Vortex! Hanger 14! BASIC Ahoy! Catacombs! Lixter! Dark Fortress! PermaLine! Starfighter! Bugout! Screens!

#38—FEB '87 Hacking into machine language utilities! Amiga RAM expanders! And ready to enter: Window Magic! Crunchman! User Conventions! The Adventurer! More BASIC 128! Jailbreak! Turtle Rescue! 640!

#39—MAR. '87 Basic esthetics! Survey of video digitizers! Multiplayer games! And ready to enter: C-64 Compressor! Wizard Tag! Turbopoke! Rescue 128! Lights Out! Pinball Arcade! Stow Away! Caverns of Geehonk!

#40—APR. '87 Inside the Amiga 2000! Fractals! Baseball games! COMAL, turtle graphics, and Logo! And ready to enter: Infoflow! Laps! Pieman! List Formatter! Scrambler! Extended Background Mode! Planet Duel!

#41—MAY '87 Kernal power! 64 and Amiga graphics! Microworlds in COMAL! Brain games! Dark Fortress master maps! And ready to enter: Moondog! Startup! Illusion Master! Wall Crawler! Scavenger Hunt!

#42—JUNE '87 Megaflops and microseconds! Sci-fi braingames! C-64 to Amiga file transfer! And ready to enter: D-Snap! Wraiths! Galactic Cab Co.! Cave of the Ice Ape! ALT-Key 128! Power Squares! 128 Multi RAM!

#43—JULY '87 Real world interfacing! Bit map graphics tutorial! C-64 graphic conversion! Martial arts software! And ready to enter: Wizard Tag II! Data Creator! Plink & Plonk! Univaders! Data Express! 128 Scroller!

#44—AUG. '87 Electronic screen swapping on the C-128! Science fiction action games! The death of GOTO! Amiga reviews! And ready to enter: Archer! Banner Print! Route 64! Steeplechase! Batter Up! Scanner!

#45—SEP. '87 The 128 versus The Clones! Building an Amiga trackball! MSD update! CES Report! And ready to enter: Crystallus! Spriteshell! Hoops! Chainmail! No News! PS to GEOS! Centerfold! Red Alert!

#46—OCT. '87 A rainbow of data structures! BASIC 8.0 reviewed! Buying guide to COMAL! Tips Ahoy! Art Gallery! And ready to enter: Empire! C-64 RAMDrive! Hotfoot! Platforms! Spray-Cam! Jam Attack!

#47—NOV '87 The ins and outs of data organization! Overview of C-64 memory expanders! Commodares! And ready to enter: Orbit! RAMCO! A-Maze-Ing! Line Sentry! Desert Front! Paper Route! Flash Flood!

#48—DEC. '87 Exploring artificial intelligence! Expansion port tutorial! Memory expanders! And ready to enter: Redirect! Silhouette! Fueling Station! Take Two! Pizza Boy! Sprite-On! Warship! Cliffhanger!

#49—JAN. '88 Iterative mathematics and graphics! Expansion port tutorial, part II! The best C-64 and Amiga games of 1987! And ready to enter: Infoflow 64! TextSave! Clone-A-Matic! Ice Titans! Jungle Jake!

#50—FEB '88 Investigating audible wave phenomena! Cartridges! Commodore 64 emulators for the Amiga! And ready to enter: Coffee Break! Crypt of Fear! Screen Wizard! ML Ranger! 128 Smart Merge! Marauder!

#51—MAR '88 Synthesizing waveforms! Comprehensive guide to Commodore power supplies! Utilizing Amiga HAM mode! Art Gallery! And ready to enter: Vee Kloros! The Extractor! 3-D Graphic Projector! Phantasy!

#52—APR. '88 BASIC's hidden treasures! Updating your Amiga 1000's printer port! Disk drive accelerators! And ready to enter: Time Subway! IRQ Messenger for the 64 and 128! Sprite Exploder! Hi-Res Windows! Barricade!

#53—MAY '88 Premier issue of *Ahoy!'s AmigaUser*, featuring: survey of sampling software! The essential Amiga game library! Guide to Amiga books! Business column! And ready to enter: AmigaUserTerm! Matrix Pattern!

#54—JUNE '88 Advanced disk drive commands! Creating your own text games, part I! Programming in COMAL! And ready to enter: Rashgar! Movie Scroll! MOB Blob! Snap Snake! Phobia! Lost Dutchman's Mine!

#55—JULY '88 Making the thermal connection! BASIC/COMAL speed tests! Text game programming, part II! And ready to enter: Tomb of Horror! Vari-Scan! Window Dressing for the 128! Guerilla! Lunar Buggy!

#56—AUG. '88 Second issue of *Ahoy!'s AmigaUser*, featuring: Video digitizers! Desktop publishing! Behind the AmigaDOS ASSIGN command! Programming hints! And ready to enter: Speech Set!

#57—SEPT. '88 Build a lightning fast compiler! Variegated COMALs! Consumer Electronics Show report! Tips Ahoy! And ready to enter: Video Snapshot! The V'Lurian Mines! Color Editor! Trap Shoot! Turbo Ski!

All issues $4.00 except #2, #3, #21 ($10.00)

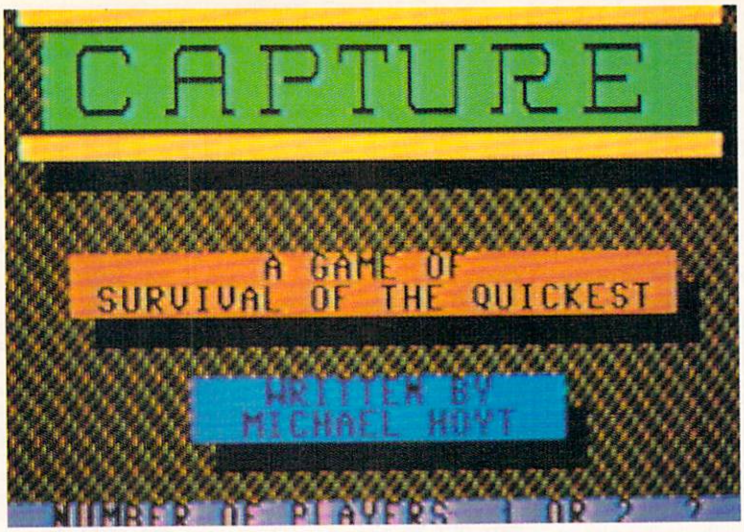
Please send me
____ copies of issue #____ ____ copies of issue #____
____ copies of issue #____ ____ copies of issue #____
Enclosed find check or money order for $________
(outside US add $1.00 per copy)

NAME________________________

ADDRESS________________________

CITY____________ STATE______ ZIP________

Send to: *Ahoy!* Back Issues, Ion International Inc., 45 West 34th Street—Suite 500, New York, NY 10001.

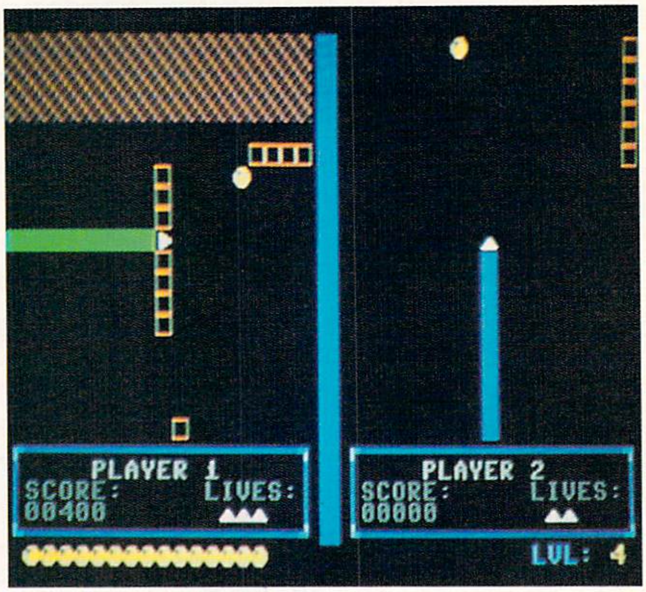


For the C-64

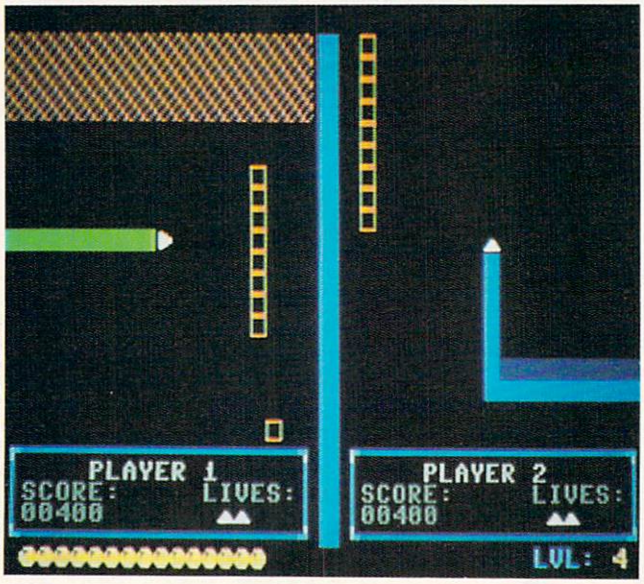
Capture is an addictive action/strategy game for the Commodore 64. The object is to achieve the highest score by capturing more objects than your opponent can. *Capture* can be played solo or head-to-head.

After saving a copy of the game to tape or disk, type RUN and press RETURN. After a few seconds a title screen will appear. Select the number of players by pressing 1 or 2. Next you will be asked for the width of the maze. Select the width using the number keys, and do the same for the maze height. Finally you will be asked for the starting level. Select a number from 1 to 8, level 1 being the least difficult. A maze will then be randomly created, and in a few seconds the game will begin.

Player 1's joystick should be plugged into Port 1, and Player 2's in Port 2. Once the game starts, you can direct the movement of your ever-growing "snake" by pushing the joystick in the desired direction. Avoid the walls and borders at all costs; colliding with one will cost you a life and give your opponent 100 points X the current level.



As you move around, you will notice that the screen shows only a small portion of the entire game board. The head of your snake is always centered on your screen, so while you are moving, the screen will scroll with you. To collect points and advance to the next level, capture all the yellow spheres in the maze. The number of spheres remaining is displayed at the bottom left of the screen. The player to retrieve the last sphere will be generously rewarded with bonus points. Note that in the one player mode you move about twice as fast as in the head-to-head mode. This faster speed makes it more difficult to move through the maze and thus partially makes up for the lack of an opponent.



PHOTOS: MICHAEL R. DAVILA

There are several tactics you can use to achieve a higher score than your opponent. The easiest method is to simply block off your opponent and cause him to crash into an object, possibly his own tail. This method will greatly increase your score, but it will also result in a shorter-lasting game, since each player begins with only three lives. A second, less difficult tactic is to save one sphere for yourself. When the other player has obtained all the other spheres, you can go back and capture the one you saved. Whoever captures the last sphere will receive all the bonus points for that level. □

SEE PROGRAM LISTING ON PAGE 67

PROGRAM LISTINGS

Attention new *Ahoy!* readers! You must read the following information very carefully prior to typing in programs listed in *Ahoy!* Certain Commodore characters, commands, and strings of characters and commands will appear in a special format. Follow the instructions and listings guide on this page.

On the following pages you'll find several programs that you can enter on your Commodore computer. But before doing so, read this entire page carefully.

To insure clear reproductions, *Ahoy!'s* program listings are generated on a daisy wheel printer, incapable of printing the commands and graphic characters used in Commodore programs. These are therefore represented by various codes enclosed in brackets [ ]. For example: the SHIFT CLR/HOME command is represented onscreen by a heart . The code we use in our listings is [CLEAR]. The chart below lists all such codes which you'll encounter in our listings, except for one other special case.

The other special case is the COMMODORE and SHIFT characters. On the front of most keys are two symbols. The symbol on the left is obtained by pressing that key while holding down the COMMODORE key; the symbol on the right, by pressing that key while holding down the SHIFT key. COMMODORE and SHIFT characters are represented in our listings by a lower-case "s" or "c" followed by the symbol of the key you must hit. COMMODORE J, for example, is represented by [c J], and SHIFT J by [s J].

Additionally, any character that occurs more than two times in a row will be displayed by a coded listing. For example, [3 "[LEFT]"] would be 3 CuRSoR left commands in a row, [5 "[s EP]"] would be 5 SHIFTed English Pounds, and so on. Multiple blank spaces will be noted in similar fashion: e.g., 22 spaces as [22 " "].

Sometimes you'll find a program line that's too long for the computer to accept (C-64 lines are a maximum of 80 characters, or 2 screen lines long; C-128 lines, a maximum of 160 characters, 2 or 4 screen lines in 40 or 80 columns respectively). To enter these lines, refer to the *BASIC Command Abbreviations Appendix* in your User Manual.

On the next page you'll find our *Bug Repellent* programs for the C-128 and C-64. The version for your machine will help you proofread programs after typing them. (Please note: the *Bug Repellent* line codes that follow each program line, in the whited-out area, should *not* be typed in. See instructions preceding each program.)

On the second page following you will find *Flankspeed*, our ML entry program, and instructions on its use.

Call *Ahoy!* at 212-239-6089 with any problems (if busy or no answer after three rings, call 212-239-0855).

WHEN YOU SEE	IT MEANS	YOU TYPE	YOU WILL SEE
[CLEAR]	Screen Clear	SHIFT CLR/HOME	
[HOME]	Home	CLR/HOME	
[UP]	Cursor Up	SHIFT ↑ CRSR ↓	
[DOWN]	Cursor Down	↑ CRSR ↓	
[LEFT]	Cursor Left	SHIFT ← CRSR →	
[RIGHT]	Cursor Right	← CRSR →	
[SS]	Shifted Space	SHIFT Space	
[INSERT]	Insert	SHIFT INST/DEL	
[DEL]	Delete	INST/DEL	
[RVSON]	Reverse On	CNTRL 9	
[RVSOFF]	Reverse Off	CNTRL 0	
[UPARROW]	Up Arrow	↑	↑
[BACKARROW]	Back Arrow	←	←
[PI]	PI	π	π
[EP]	English Pound	£	£

WHEN YOU SEE	IT MEANS	YOU TYPE	YOU WILL SEE
[BLACK]	Black	CNTRL 1	
[WHITE]	White	CNTRL 2	
[RED]	Red	CNTRL 3	
[CYAN]	Cyan	CNTRL 4	
[PURPLE]	Purple	CNTRL 5	
[GREEN]	Green	CNTRL 6	
[BLUE]	Blue	CNTRL 7	
[YELLOW]	Yellow	CNTRL 8	
[F1]	Function 1	F1	
[F2]	Function 2	SHIFT F1	
[F3]	Function 3	F3	
[F4]	Function 4	SHIFT F3	
[F5]	Function 5	F5	
[F6]	Function 6	SHIFT F5	
[F7]	Function 7	F7	
[F8]	Function 8	SHIFT F7	

BUG REPELLENT FOR THE 64 & 128 By BUCK CHILDRESS

Please note: the *Bug Repellent* programs listed here are for *Ahoy!* programs published from the May 1987 issue onward! For older programs, use the older version.

Type in, save, and run *Bug Repellent.* You'll be asked if you want automatic saves to take place. If so, you're prompted for the device, DISK (D) or TAPE (T). You then pick a starting file number, 0 through 99. Next, you enter a name, up to 14 characters long. At this point, *Bug Repellent* verifies your entries and gives you a chance to change them if you want. If no changes are needed, *Bug Repellent* activates itself. (Pressing RETURN without answering the prompts defaults to disk drive and begins your files with "00BACKUP".)

Type NEW and begin entering an *Ahoy!* program. As you enter program lines and press RETURN, a *Bug Repellent* code appears at the top of your screen. If it doesn't match the code in the program listing, an error exists. Correct the line and the codes will match.

If used, automatic saves take place every 15 minutes. When the RETURN key is pressed on a program line, the screen changes color to let you know that a save will begin in about three seconds. You may cancel the save by pressing the RUN STOP key. The file number increments after each save. It resets to 00 if 99 is surpassed. After saving, or cancelling, the screen returns to its original color and the timer resets for 15 minutes.

When you've finished using *Bug Repellent*, deactivate it by typing SYS 49152 [RETURN] for the Commodore 64 or SYS 4864 [RETURN] for the Commodore 128.

C-64 BUG REPELLENT

```
•10 PRINTCHR$(147)"LOADING AND CHECKING THE DATA[3"."]":J
=49152
•20 FORB=0TO11:READA:IFA<0ORA>255THEN40
•30 POKEJ+B,A:X=X+A:NEXTB:READA:IFA=XTHEN50
•40 PRINT:PRINT"ERROR IN DATA LINE:"PEEK(64)*256+PEEK(63)
:END
•50 X=0:J=J+12:IFJ<49456THEN20
•60 POKE198,0:POKE49456,0:A$="Y":B$=A$:C$="D":D$="DISK":D
=8:PRINTCHR$(147)
•70 INPUT"DO YOU WANT AUTOMATIC SAVES (Y/N)";A$:PRINT:IFA
$="Y"THEN90
•80 PRINT"NO AUTOMATIC SAVES[3"."]":GOTO150
•90 POKE49456,1:INPUT"DISK OR TAPE (D/T)";C$:IFC$<>"D"THE
ND=1:D$="TAPE"
•100 POKE49457,D:D$=D$+" DRIVE":PRINT:INPUT"FILE NUMBER (
0-99)";N
•110 N$=RIGHT$(STR$(N),2):IFN<10THENN$=CHR$(48)+CHR$(N+48
)
•120 F$="BACKUP":PRINT:INPUT"FILENAME";F$:F$=N$+LEFT$(F$,
14):L=LEN(F$)
•130 POKE49458,L:FORJ=1TOL:POKE49458+J,ASC(MID$(F$,J,1)):
NEXTJ:PRINT
•140 PRINT"SAVING DEVICE ** "D$:PRINT"STARTING WITH ** "F
$
•150 PRINT:INPUT"IS THIS CORRECT (Y/N)";B$:IFB$<>"Y"THEN6
0
•160 POKE770,131:POKE771,164:SYS49152:END
•170 DATA169,79,32,210,255,162,38,160,192,204,3,3,1507
•180 DATA208,10,162,131,160,164,169,70,32,210,255,44,1615
•190 DATA169,78,32,210,255,142,2,3,140,3,3,76,1113
•200 DATA36,193,32,96,165,134,122,132,123,32,115,0,1180
•210 DATA170,240,243,162,255,134,58,144,3,76,150,164,1799
•220 DATA32,107,169,32,121,165,173,0,2,240,5,169,1215
•230 DATA79,141,2,3,76,162,164,169,0,133,2,133,1064
•240 DATA251,133,252,133,254,24,101,20,69,254,230,254,197
5
•250 DATA24,101,21,69,254,170,230,254,164,252,185,0,1724
•260 DATA2,133,253,201,34,208,6,165,2,73,255,133,1465
•270 DATA2,201,32,208,4,165,2,240,8,138,24,101,1125
•280 DATA253,69,254,170,44,198,254,230,252,164,253,208,23
49
•290 DATA213,138,41,240,74,74,74,74,24,105,129,141,1327
•300 DATA44,193,138,41,15,24,105,129,141,45,193,162,1230
•310 DATA0,189,43,193,240,12,157,0,4,173,134,2,1147
•320 DATA157,0,216,232,208,239,169,38,141,2,3,173,1578
•330 DATA48,193,240,23,165,161,201,212,176,4,165,160,1748
•340 DATA240,13,238,32,208,160,0,32,225,255,208,6,1617
•350 DATA32,33,193,76,38,192,232,208,242,200,208,239,1893
•360 DATA32,68,229,169,0,168,174,49,193,32,186,255,1555
•370 DATA173,50,193,162,51,160,193,32,189,255,169,43,1670
•380 DATA166,45,164,46,32,216,255,162,1,189,51,193,1520
•390 DATA168,200,152,201,58,144,2,169,48,157,51,193,1543
•400 DATA201,48,208,3,202,16,234,32,33,193,76,116,1362
•410 DATA164,206,32,208,169,0,170,168,76,219,255,160,1827
•420 DATA1,1,160,0,0,65,72,79,89,33,0,0,500
```

C-128 BUG REPELLENT

```
•10 PRINTCHR$(147)"LOADING AND CHECKING THE DATA[3"."]":J
=4864
•20 FORB=0TO11:READA:IFA<0ORA>255THEN40
•30 POKEJ+B,A:X=X+A:NEXTB:READA:IFA=XTHEN50
•40 PRINT:PRINT"ERROR IN DATA LINE:"PEEK(66)*256+PEEK(65)
:END
•50 X=0:J=J+12:IFJ<5213THEN20
•60 POKE208,0:POKE5213,0:A$="Y":B$=A$:C$="D":D$="DISK":D=
8:PRINTCHR$(147)
•70 INPUT"DO YOU WANT AUTOMATIC SAVES (Y/N)";A$:PRINT:IFA
$="Y"THEN90
•80 PRINT"NO AUTOMATIC SAVES[3"."]":GOTO150
•90 POKE5213,1:INPUT"DISK OR TAPE (D/T)";C$:IFC$<>"D"THEN
D=1:D$="TAPE"
•100 POKE5214,D:D$=D$+" DRIVE":PRINT:INPUT"FILE NUMBER (0
-99)";N
•110 N$=RIGHT$(STR$(N),2):IFN<10THENN$=CHR$(48)+CHR$(N+48
)
•120 F$="BACKUP":PRINT:INPUT"FILENAME";F$:F$=N$+LEFT$(F$,
14):L=LEN(F$)
•130 POKE5215,L:FORJ=1TOL:POKE5215+J,ASC(MID$(F$,J,1)):NE
XTJ:PRINT
•140 PRINT"SAVING DEVICE ** "D$:PRINT"STARTING WITH ** "F
$
•150 PRINT:INPUT"IS THIS CORRECT (Y/N)";B$:IFB$<>"Y"THEN6
0
•160 POKE770,198:POKE771,77:SYS4864:END
•170 DATA32,58,20,169,41,162,19,236,3,3,208,4,955
•180 DATA169,198,162,77,141,2,3,142,3,3,224,19,1143
•190 DATA208,7,32,125,255,79,78,0,96,32,125,255,1292
•200 DATA79,70,70,0,96,162,0,134,251,189,0,2,1053
•210 DATA240,19,201,48,144,9,201,58,176,5,133,251,1485
•220 DATA232,208,238,134,252,165,251,208,3,76,198,77,2042
•230 DATA169,0,166,235,164,236,133,253,133,254,142,47,193
2
•240 DATA20,140,48,20,24,101,22,69,254,230,254,24,1206
•250 DATA101,23,69,254,170,230,254,164,252,185,0,2,1704
•260 DATA133,251,201,34,208,6,165,253,73,255,133,253,1965
•270 DATA201,32,208,4,165,253,240,8,138,24,101,251,1625
•280 DATA69,254,170,44,198,254,230,252,164,251,208,213,23
07
•290 DATA138,41,240,74,74,74,74,24,105,65,141,88,1138
•300 DATA20,138,41,15,24,105,65,141,89,20,32,79,769
•310 DATA20,189,85,20,240,6,32,210,255,232,208,245,1742
•320 DATA174,47,20,172,48,20,24,32,240,255,173,93,1298
•330 DATA20,240,27,165,161,201,212,176,4,165,160,240,1771
•340 DATA17,32,65,20,238,32,208,238,1,214,32,225,1322
•350 DATA255,208,6,32,49,20,76,198,77,232,208,242,1603
•360 DATA200,208,239,32,66,193,173,95,20,162,96,160,1644
•370 DATA20,32,189,255,169,0,170,32,104,255,169,0,1395
•380 DATA174,94,20,168,32,186,255,169,45,174,16,18,1351
•390 DATA172,17,18,32,216,255,162,1,189,96,20,168,1346
•400 DATA200,152,201,58,144,2,169,48,157,96,20,201,1448
•410 DATA48,208,3,202,16,234,32,49,20,141,0,2,955
•420 DATA76,183,77,58,59,32,65,20,206,32,208,206,1222
•430 DATA1,214,169,0,170,168,76,219,255,32,79,20,1403
•440 DATA169,26,141,0,214,173,0,214,16,251,96,162,1462
•450 DATA0,142,0,255,96,19,18,32,32,32,32,146,804
•460 DATA0,1,0,0,65,72,79,89,33,0,0,0,339
```

FLANKSPEED FOR THE C-64 By GORDON F. WHEAT

Flankspeed will allow you to enter machine language *Ahoy!* programs without any mistakes. Once you have typed the program in, save it for future use. While entering an ML program with *Flankspeed* there is no need to enter spaces or hit the carriage return. This is all done automatically. If you make an error in a line a bell will ring and you will be asked to enter it again. To LOAD in a program Saved with *Flankspeed* use LOAD "name",1,1 for tape, or LOAD "name",8,1 for disk. The function keys may be used after the starting and ending addresses have been entered.

f1–SAVEs what you have entered so far.

f3–LOADs in a program worked on previously.

f5–To continue on a line you stopped on after LOADing in the previous saved work.

f7–Scans through the program to locate a particular line, or to find out where you stopped the last time you entered the program. It temporarily freezes the output as well.

```
·100 POKE53280,12:POKE53281,11                          OP
·105 PRINT"[CLEAR][c 8][RVSON][15" "]FLANKSPEED[15" "]";  FP
·110 PRINT"[RVSON][5" "]MISTAKEPROOF ML ENTRY PROGRAM[6" "
 ]"                                                      JP
·115 PRINT"[RVSON][9" "]CREATED BY G. F. WHEAT[9" "]"    FA
·120 PRINT"[RVSON][3" "]COPR. 1987, ION INTERNATIONAL INC.
 [3" "]"                                                 AJ
·125 FORA=54272TO54296:POKEA,0:NEXT                      ND
·130 POKE54272,4:POKE54273,48:POKE54277,0:POKE54278,249:PO
 KE54296,15                                              NP
·135 FORA=680TO699:READB:POKEA,B:NEXT                    FL
·140 DATA169,251,166,253,164,254,32,216,255,96           FF
·145 DATA169,0,166,251,164,252,32,213,255,96             EK
·150 B$="STARTING ADDRESS IN HEX":GOSUB430:AD=B:SR=B     KP
·155 GOSUB480:IFB=0THEN150                               OE
·160 POKE251,T(4)+T(3)*16:POKE252,T(2)+T(1)*16           AM
·165 B$="ENDING ADDRESS IN HEX":GOSUB430:EN=B            PE
·170 GOSUB470:IFB=0THEN150                               PG
·175 POKE254,T(2)+T(1)*16:B=T(4)+1+T(3)*16               GM
·180 IFB>255THENB=B-255:POKE254,PEEK(254)+1              HG
·185 POKE253,B:PRINT                                     EC
·190 REM  GET HEX LINE                                   ED
·195 GOSUB495:PRINT": [c P][LEFT]";:FORA=0TO8            KD
·200 FORB=0TO1:GOTO250                                   IH
·205 NEXTB                                               IJ
·210 A%(A)=T(1)+T(0)*16:IFAD+A-1=ENTHEN340               FA
·215 PRINT" [c P][LEFT]";                                EG
·220 NEXTA:T=AD-(INT(AD/256)*256):PRINT" "               II
·225 FORA=0TO7:T=T+A%(A):IFT>255THENT=T-255              GL
·230 NEXT                                                GI
·235 IFA%(8)<>TTHENGOSUB375:GOTO195                      FL
·240 FORA=0TO7:POKEAD+A,A%(A):NEXT:AD=AD+8:GOTO195       IM
·245 REM    GET HEX INPUT                                PA
·250 GETA$:IFA$=""THEN250                                GA
·255 IFA$=CHR$(20)THEN305                                GO
·260 IFA$=CHR$(133)THEN535                               LM
·265 IFA$=CHR$(134)THEN560                               IG
·270 IFA$=CHR$(135)THENPRINT" ":GOTO620                  HO
·275 IFA$=CHR$(136)THENPRINT" ":GOTO635                  HE
·280 IFA$>"@"ANDA$<"G"THENT(B)=ASC(A$)-55:GOTO295        MI
·285 IFA$>"/"ANDA$<":"THENT(B)=ASC(A$)-48:GOTO295        DJ
·290 GOSUB415:GOTO250                                    JA
·295 PRINTA$"[c P][LEFT]";                               PK
·300 GOTO205                                             FA
·305 IFA>0THEN320                                        BI
·310 A=-1:IFB=1THEN330                                   BB
·315 GOTO220                                             FA
·320 IFB=0THENPRINTCHR$(20);CHR$(20);:A=A-1              BF
·325 A=A-1                                               FK
·330 PRINTCHR$(20);:GOTO220                              PH
·335 REM LAST LINE                                       CP
·340 PRINT" ":T=AD-(INT(AD/256)*256)                     KH
·345 FORB=0TOA-1:T=T+A%(B):IFT>255THENT=T-255            OD
·350 NEXT                                                OB
·355 IFA%(A)<>TTHENGOSUB375:GOTO195                      LH
·360 FORB=0TOA-1:POKEAD+B,A%(B):NEXT                     BO
·365 PRINT:PRINT"YOU ARE FINISHED!":GOTO535              MB
·370 REM BELL AND ERROR MESSAGES                         LM
·375 PRINT:PRINT"LINE ENTERED INCORRECTLY":PRINT:GOTO415 JK
·380 PRINT:PRINT"INPUT A 4 DIGIT HEX VALUE!":GOTO415     PG
·385 PRINT:PRINT"ENDING IS LESS THAN STARTING!":B=0:GOTO41
 5                                                       IK
·390 PRINT:PRINT"ADDRESS NOT WITHIN SPECIFIED RANGE!":B=0:
 GOTO415                                                 HK
·395 PRINT:PRINT"NOT ZERO PAGE OR ROM!":B=0:GOTO415      DM
·400 PRINT"?ERROR IN SAVE":GOTO415                       JK
·405 PRINT"?ERROR IN LOAD":GOTO415                       IO
·410 PRINT:PRINT:PRINT"END OF ML AREA":PRINT             JO
·415 POKE54276,17:POKE54276,16:RETURN                    BF
·420 OPEN15,8,15:INPUT#15,A,A$:CLOSE15:PRINTA$:RETURN    DH
·425 REM GET FOUR DIGIT HEX                              IM
·430 PRINT:PRINTB$;:INPUTT$                              OL
·435 IFLEN(T$)<>4THENGOSUB380:GOTO430                    JD
·440 FORA=1TO4:A$=MID$(T$,A,1):GOSUB450:IFT(A)=16THENGOSUB
 380:GOTO430                                             AK
·445 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)*16)+T(4):RETURN KB
·450 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-55:RETURN         GM
·455 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-48:RETURN         NJ
·460 T(A)=16:RETURN                                      IC
·465 REM ADDRESS CHECK                                   OL
·470 IFAD>ENTHEN385                                      HO
·475 IFB<SRORB>ENTHEN390                                 LE
·480 IFB<256OR(B>40960ANDB<49152)ORB>53247THEN395        OB
·485 RETURN                                              HE
·490 REM ADDRESS TO HEX                                  PM
·495 AC=AD:A=4096:GOSUB520                               AP
·500 A=256:GOSUB520                                      NF
·505 A=16:GOSUB520                                       LG
·510 A=1:GOSUB520                                        HE
·515 RETURN                                              JD
·520 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):GOTO530          OC
·525 A$=CHR$(T+48)                                       JI
·530 PRINTA$;:AC=AC-A*T:RETURN                           AA
·535 A$="**SAVE**":GOSUB585                              IC
·540 OPEN1,T,1,A$:SYS680:CLOSE1                          AB
·545 IFST=0THENEND                                       FB
·550 GOSUB400:IFT=8THENGOSUB420                          PM
·555 GOTO535                                             FI
·560 A$="**LOAD**":GOSUB585                              PE
·565 OPEN1,T,0,A$:SYS690:CLOSE1                          PO
·570 IFST=64THEN195                                      OI
·575 GOSUB405:IFT=8THENGOSUB420                          CO
·580 GOTO560                                             GN
·585 PRINT" ":PRINTTAB(14)A$                             KA
·590 PRINT:A$="":INPUT"FILENAME";A$                      IO
·595 IFA$=""THEN590                                      HK
·600 PRINT:PRINT"TAPE OR DISK?":PRINT                    HL
·605 GETB$:T=1:IFB$="D"THENT=8:A$="@0:"+A$:RETURN        NP
·610 IFB$<>"T"THEN605                                    KO
·615 RETURN                                              PH
·620 B$="CONTINUE FROM ADDRESS":GOSUB430:AD=B            DD
·625 GOSUB475:IFB=0THEN620                               NK
·630 PRINT:GOTO195                                       HN
·635 B$="BEGIN SCAN AT ADDRESS":GOSUB430:AD=B            FK
·640 GOSUB475:IFB=0THEN635                               LN
·645 PRINT:GOTO670                                       HI
·650 FORB=0TO7:AC=PEEK(AD+B):GOSUB505:IFAD+B=ENTHENAD=SR:G
 OSUB410:GOTO195                                         LM
·655 PRINT" ";:NEXTB                                     LE
·660 PRINT:AD=AD+8                                       CD
·665 GETB$:IFB$=CHR$(136)THEN195                         JD
·670 GOSUB495:PRINT": ";:GOTO650                         KE
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 59 and 60 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

BASIC METAMORPHOSIS FROM PAGE 32

MINI-COMP

```
·1 REM==================================  IH
·2 REM          MINI-COMP                  BA
·3 REM     RUPERT REPORT #58               OL
·4 REM A MINIMAL COMPILER FOR THE C-64     IN
·5 REM==================================  ML
·6 REM     RUN 2000 TO COMPILE             BE
·7 REM==================================  FF
·8 REM                                     JJ
·9 REM   LINES 1 THROUGH 999 ARE EXAMPLES
  OF ALL STATEMENTS ALLOWED BY MINI-COMP   GP
·10 A=50                                   GP
·20 B=-10                                  AI
·30 C=A                                    FD
·40 D=A+B                                  FH
·50 IF A=B THEN 60                         AE
·60 GOTO 70                                CJ
·70 PRINT                                  OB
·80 PRINT A                                CO
·90 PRINT B;                               HH
·100 PRINT CHR$(C)                         FC
·110 PRINT CHR$(D);                        LO
·999 END                                   GK
·1000 REM --- COMMON ROUTINES ---          NM
·1100 C=PEEK(M): M=M+1: PRINT C,: IF C=32
 THEN 1100: REM IGNORE SPACES              HG
·1110 IF C=0 THEN PRINT                    EO
·1120 RETURN                               PC
·1300 VF=TRUE: IF C<65 OR C>90 THEN VF=FA
LSE: RETURN                                IK
·1310 AD=(C-65)*2 + VM                     CF
·1320 NX=AD: GOSUB 1400: A0=NL: A1=NH   :
 REM VAR LSB                               BP
·1330 NX=AD+1: GOSUB 1400: A2=NL: A3=NH :
 REM VAR MSB                               MA
·1340 RETURN                               MP
·1400 NH=INT(NX/256)                       KK
·1410 NL=NX-256*NH                         AC
·1420 RETURN                               BP
·1500 FOR KK=1 TO N                        MG
·1510 IF CM>EM THEN PRINT"OUT OF MEMORY -
 COMPILED PRGM TOO LARGE": END             ED
·1520 POKE CM,C(KK)                        FK
·1530 PRINT CM;":";C(KK)                   EC
·1540 CM=CM+1: NEXT                        HN
·1550 RETURN                               KC
·2000 REM ====== INITIALIZATION ========= PB
·2010 DIM LL(50,2)        :REM LL(N,1)=LINE
 # OF NTH LINE                             ON
·2015 :REM  LL(N,2)=COMPILED MEM LOCATION
 OF THIS LINE                              DC
·2020 DIM C(50)    :REM STORES OBJECT CODE
 BYTES                                     BI
·2030 FALSE=0: TRUE=NOT FALSE              IG
·2040 DEF FNPTR(M)=PEEK(M)+256*PEEK(M+1)   EH
·2050 VM=49152   :REM $C000 START OF VARIA
BLE MEM                                    AG
·2060 PM=49408   :REM $C100 START OF OBJEC
T MEMORY                                   LM
·2070 EM=53247   :REM $CFFF END OF OBJECT
MEMORY                                     BK
·2080 BT=2049    :REM $0800 START OF BASIC
 TEXT                                      MC
·2085 GOSUB 11000    :REM PUT PRINT RTN IN
 MEM                                       DI
·2090 M=BT          :REM NEXT SOURCE MEMORY
 TO PEEK                                   BA
·2100 SN=1      :REM CURRENT SOURCE STATE
MENT NUMBER                                DB
·2110 CM=PM         :REM NEXT OBJECT MEMORY
 TO POKE                                   NL
·2120 FOR N=VM TO VM+51: POKE N,0: NEXT :
 REM CLR VAR'S                             EM
·2125 REM ============ MAIN =========== MI
·2130 PTR=FNPTR(M): M=M+2    :REM NEXT LIN
E PTR                                      LJ
·2140 LN=FNPTR(M): M=M+2      :REM CURRENT
 LINE #                                    IC
·2150 IF LN>999 THEN PRINT "[3"="] END OF
 PASS 1 [3"="]": GOTO 2400                 BE
·2160 PRINT "[5" "]CURRENT LINE # ="; LN  HC
·2170 LL(SN,1)=LN      :REM CURRENT LINE # EG
·2180 LL(SN,2)=CM             :REM START OBJ
 MEM LOC                                   GC
·2190 SN=SN+1    :REM # SOURCE STATEMENTS AG
·2200 REM --- GET BYTE ---                 FP
·2210 GOSUB 1100     :REM FETCH NEXT BYTE  EO
·2220 GOSUB 1300: IF VF THEN GOSUB 3000 :
 GOTO 2290: REM 'VARIABLE                  BE
·2230 IF C=139 THEN GOSUB 4000: GOTO 2290
 : REM 'IF                                 PL
·2240 IF C=137 THEN GOSUB 5000: GOTO 2290
 : REM 'GOTO                               LN
·2250 IF C=153 THEN GOSUB 6000: GOTO 2290
 : REM 'PRINT                              BE
·2260 IF C=128 THEN GOSUB 7000: GOTO 2290
 : REM 'END                                BA
·2270 IF C=143 THEN GOSUB 8000: GOTO 2290
 : REM 'REM                                AB
·2280 PRINT "UNKNOWN COMMAND CODE";C;"IN
LINE";LL(SN,1):STOP                        IP
·2290 IF C>0 THEN EC=0: GOSUB 10000: STOP
 :REM 'EOL                                 CH
·2300 GET K$:IF K$="" THEN 2130 :REM BACK
 FOR MORE                                  OA
·2310 GET K$:IF K$="" THEN 2310            LD
·2320 GOTO 2130                            GC
·2400 REM  - PASS 2 - FIX JUMP ADDRESSES  DN
```

```
•2410 IF JI=0 THEN 2570 :REM NO JUMPS     OO
•2420 FOR N=1 TO JI   :REM CHECK ITEMS IN
 JUMP TABLE                               PG
•2430 :MM=JT(N,1) :REM REFERENCED LINE #  AD
•2440 :FOR J=1 TO SN    :REM CHECK ACTUAL
 LINE #S                                  AA
•2445 :REM   - GET OBJ MEM TARGET ADDRESS
 AND JMP ADDRESS:                         NE
•2450 :IF MM=LL(J,1) THEN TADDR=JT(N,2):
 JADDR=LL(J,2): GOTO 2490                 JH
•2460 :NEXT J                             KG
•2470 :REM NO MATCH FOUND                 OI
•2480 :EC=1: LN=JT(N,0): GOSUB 10000:STOP HB
•2490 :NX=JADDR :REM ADDR OF LINE # MM    CB
•2530 :GOSUB 1400 :REM CONVERT LINE #     KB
•2540 :POKE TADDR,NL      :REM USE ADDR IN
 JUMP TABLE                               EP
•2550 :POKE TADDR+1,NH                    NL
•2560 NEXT N     :REM NEXT JUMP TABLE ITEM JJ
•2570 PRINT"[3"="] END OF PASS 2 [3"="]"  ON
•2580 PRINT"TO EXECUTE THE COMPILED PROGR
AM, ENTER"                                HP
•2590 PRINT" SYS"; PM                     AA
•2595 PRINT"OBJECT CODE RESIDES FROM";PM;
"TO";CM-1                                 OJ
•2600 END                                 LC
•3000 REM <<<  A=(-)NN, A=B, A=B+C >>>    GL
•3010 D0=A0: D1=A1 :REM ADDR OF A'S LSB   GE
•3020 D2=A2: D3=A3 :REM A'S MSB           HA
•3030 GOSUB 1100: IF C<>178 THEN EC=178:
EC$=" = ": GOSUB 10000: STOP    :REM '=   AA
•3040 GOSUB 1100    :REM '-, NN, OR B     DJ
•3050 GOSUB 1300: IF VF THEN 3400 :REM 'B EL
•3055 REM <<< A=(-)NN >>>                 LP
•3060 IF C=171 THEN C$="-" :REM '-        HM
•3065 IF C<>171 THEN C$=" "+CHR$(C) :REM
'0-9                                      DI
•3070 GOSUB 1100  :REM GET DIGITS OF NN   FN
•3080 IF C=0 THEN 3120                    DB
•3085 IF CHR$(C)<"0" OR CHR$(C)>"9" THEN
EC=48: GOSUB 10000: STOP                  LM
•3090 C$=C$+CHR$(C)                       LJ
•3100 GOTO 3070                           HP
•3120 NN=VAL(C$)                          OO
•3130 NN%=NN    :REM ERROR CHECK          BO
•3140 IF NN<0 THEN NN=NN+65536       :REM
 CONVERT (-32768,32767) TO (0,65535)      HG
•3150 NX=NN: GOSUB 1400                   NH
•3170 MSB=NH: LSB=NL                      AB
•3175 :REM LDA #NN(LSB), STA A(LSB), LDA
 #NN(MSB), STA A(MSB)                     EG
•3180 N=10: C(1)=169: C(2)=LSB: C(3)=141:
 C(4)=D0: C(5)=D1                         NP
•3190 C(6)=169: C(7)=MSB: C(8)=141: C(9)=
D2: C(10)=D3                              CN
•3200 GOSUB 1500     :REM POKE VALUES INTO
 OBJECT MEMORY                            HF
•3210 RETURN                              CE
```

```
•3400 REM <<< A=B OR A=B+C >>>            BN
•3410 S0=A0: S1=A1    :REM B'S LSB ADDR   PN
•3420 S2=A2: S3=A3    :REM B'S MSB        AL
•3430 GOSUB 1100                          LO
•3440 IF C=0 THEN 3600    :REM A=B        DM
•3450 REM <<< A=B+C >>>                   KF
•3460 IF C<>170 THEN EC=170:  EC$=" + ":
GOSUB 10000: STOP :REM TEST '+            HG
•3470 GOSUB 1100: GOSUB 1300: IF NOT VF T
HEN EC=65: GOSUB 10000: STOP: REM 'C      DK
•3480 GOSUB 1100: IF C>0 THEN EC=0: GOSUB
 10000: STOP :REM 'EOL                    ME
•3490 S4=A0: S5=A1    :REM ADDR C'S LSB   OI
•3500 S6=A2: S7=A3    :REM C'S MSB        NE
•3530 REM CLC, LDA B(LSB), ADC C(LSB),STA
 A(LSB)                                   DJ
•3535 REM LDA B(MSB), ADC C(MSB), STA A(M
SB)                                       MP
•3540 N=19: C(1)=24: C(2)=173: C(3)=S0: C
(4)=S1                                    LD
•3550 C(5)=109: C(6)=S4: C(7)=S5: C(8)=14
1: C(9)=D0: C(10)=D1                      HD
•3560 C(11)=173: C(12)=S2: C(13)=S3: C(14
)=109: C(15)=S6: C(16)=S7                 PK
•3570 C(17)=141: C(18)=D2: C(19)=D3       DI
•3580 GOSUB 1500                          GA
•3590 RETURN                              KC
•3600 REM <<< A=B >>>                     MJ
•3610 REM LDA B(LSB), STA A(LSB), LDA B(M
SB), STA A(MSB)                           MG
•3620 N=12: C(1)=173: C(2)=S0: C(3)=S1: C
(4)=141: C(5)=D0: C(6)=D1                 IB
•3630 C(7)=173: C(8)=S2: C(9)=S3: C(10)=1
41: C(11)=D2: C(12)=D3                    GC
•3640 GOSUB 1500                          JJ
•3650 RETURN                              NO
•4000 REM <<<  IF A=B THEN MM  >>>        KG
•4010 GOSUB 1100  :REM 'A                 EL
•4020 GOSUB 1300: IF NOT VF THEN EC=65:
 GOSUB 10000: STOP                        KP
•4030 S0=A0: S1=A1: S2=A2: S3=A3          OK
•4040 GOSUB 1100: IF C<>178 THEN EC=178:
 GOSUB 10000: STOP   :REM '=              IK
•4050 GOSUB 1100       :REM 'B            FC
•4060 GOSUB 1300: IF NOT VF THEN EC=65:
 GOSUB 10000: STOP                        JH
•4070 GOSUB 1100       :REM 'THEN         FE
•4080 IF C<>167 THEN EC=167: GOSUB 10000:
 STOP                                     GL
•4090 C$=""                               GE
•4100 GOSUB 1100: IF C=0 THEN 4120   :REM
 GET MM                                   EI
•4110 C$=C$+CHR$(C): GOTO 4100            BA
•4120 MM=VAL(C$)                          OI
•4130 IF MM<0 OR MM>999 THEN EC=1:  GOSUB
 10000: STOP                              HK
•4140 JI=JI+1        :REM JUMP TABLE INDEX OL
•4150 JT(JI,0)=LN    :REM CURRENT LINE #  ML
```

```
•4160 JT(JI,1)=MM    :REM TARGET LINE #      EB
•4170 JT(JI,2)=CM+17    :REM MEM LOC AFTER
 'JMP                                        FB
•4180 REM  LDA A(MSB), CMP B(MSB), BNE EX
IT,                                          OC
•4185 REM  LDA A(LSB), CMP B(LSB), BNE EX
IT, JMP MM                                   KA
•4190 N=19: C(1)=173: C(2)=S2: C(3)=S3       GK
•4200 C(4)=205: C(5)=A2: C(6)=A3             JG
•4210 C(7)=208: C(8)=11: C(9)=173: C(10)=
S0: C(11)=S1                                 CE
•4220 C(12)=205: C(13)=A0: C(14)=A1          GB
•4230 C(15)=208: C(16)=3: C(17)=76: C(18)
=0: C(19)=0                                  OA
•4240 GOSUB 1500                             PD
•4250 RETURN                                 DI
•5000 REM <<< GOTO MM >>>                    GB
•5010 C$=""                                  AA
•5020 GOSUB 1100: REM GET DIGITS OF MM       IB
•5030 IF C=0 THEN 5060                       MP
•5040 C$=C$+CHR$(C)                          HG
•5050 GOTO 5020                              BJ
•5060 MM=VAL(C$)                             IN
•5070 IF MM<0 OR MM>999 THEN EC=1:  GOSUB
 10000: STOP                                 NJ
•5080 JI=JI+1       :REM JUMP TABLE INDEX    IA
•5085 JT(JI,0)=LN   :REM SOURCE LINE#        CG
•5090 JT(JI,1)=MM   :REM TARGET LINE#        MM
•5100 JT(JI,2)=CM+1 :REM OBJ MEM LOCATION
 AFTER 'JMP'                                 NC
•5110 REM  JMP MM                            BP
•5120 N=3: C(1)=76: C(2)=0: C(3)=0           FO
•5130 GOSUB 1500                             HB
•5140 RETURN                                 LG
•6000 REM <<< PRINT, PRINT A[;], OR PRINT
 CHR$(A)[;] >>>                              NH
•6010 GOSUB 1100: IF C=199 THEN 6300 :REM
 'CHR$                                       NL
•6020 IF C=0 THEN PC=13: GOSUB 6200: RETU
RN:  REM 'PRINT                              KP
•6030 GOSUB 1300: IF NOT VF THEN EC=65:
 GOSUB 10000: STOP                           EB
•6040 REM <<< PRINT A >>>                    HK
•6050 REM   LDX $A(MSB), LDY $A(LSB), JSR
 $C0E0                                       FM
•6060 PRINT: N=9: C(1)=174: C(2)=A0: C(3)
=A1                                          AK
•6070 C(4)=172: C(5)=A2: C(6)=A3             BA
•6080 C(7)=32: C(8)=224: C(9)=192            DE
•6090 GOSUB 1500                             DI
•6095 PC=32: GOSUB 6200     :REM ADD SPACE
 AFTER  DIGITS                               KK
•6100 GOSUB 1100: IF C=0 THEN PC=13: GOSU
B 6200: RETURN    :REM PRINT CR              PO
•6110 IF C<>59 THEN EC=59: GOSUB 10000:
 STOP  :REM  ' ;                             KL
•6120 GOSUB 1100: IF C>0 THEN EC=0: GOSUB
 10000: STOP                                 EN
•6130 RETURN                                 JH
•6200 REM --- PRINT CHARACTER PC ---         HF
•6210 REM LDA #PC, JSR $FFD2                 JI
•6220 PRINT:N=5: C(1)=169: C(2)=PC: C(3)=
32                                           PG
•6230 C(4)=210: C(5)=255: GOSUB 1500         DG
•6240 RETURN                                 AG
•6300 REM <<< PRINT CHR$(A) [;] >>>          MA
•6310 GOSUB 1100: IF C<>40 THEN EC=40: GO
SUB 10000: STOP :REM  '(                     HP
•6320 GOSUB 1100: GOSUB 1300: IF NOT VF T
HEN EC=65: GOSUB 10000: STOP                 BE
•6330 REM    LDA A(LSB), JSR $FFD2           OL
•6340 PRINT: N=6: C(1)=173: C(2)=A0: C(3)
=A1                                          HD
•6350 C(4)=32: C(5)=210: C(6)=255            NP
•6360 GOSUB 1500                             ED
•6370 GOSUB 1100: IF C<>41 THEN EC=41: GO
SUB 10000: STOP  : REM ')                    NO
•6380 GOSUB 1100: IF C>0 THEN 6400           JO
•6390 PRINT: PC=13: GOSUB 6200: RETURN       NG
•6400 IF C<>59 THEN EC=59: GOSUB 10000:
 STOP  :REM ';                               HP
•6410 GOSUB 1100: IF C<>0 THEN EC=0:GOSUB
 10000: STOP                                 PP
•6420 RETURN                                 LL
•7000 REM <<< END >>>                        EL
•7005 REM    RTS                             AJ
•7010 N=1: C(1)=96: GOSUB 1500               IH
•7020 GOSUB 1100: IF C<>0 THEN EC=0:GOSUB
 10000: STOP                                 DD
•7030 RETURN                                 BP
•8000 REM <<< REM >>>                        DE
•8010 GOSUB 1100: IF C>0 THEN 8010           PA
•8020 RETURN                                 AB
•10000 PRINT:PRINT"[RVSON]SYNTAX ERROR IN
 LINE[RVSOFF]";LN                            MC
•10010 IF EC=0 THEN PRINT"EXPECTED END-OF
 -LINE NOT FOUND":RETURN                     AN
•10020 IF EC=1 THEN PRINT"INVALID LINE NU
MBER":RETURN                                 AJ
•10030 IF EC=48 THEN PRINT"NUMERIC VALUE
 0-9 EXPECTED": RETURN                       DO
•10040 IF EC=65 THEN PRINT"VARIABLE A-Z E
XPECTED": RETURN                             PD
•10050 IF EC>127 THEN PRINT"EXPECTED BASI
C KEYWORD ";EC$: RETURN                      NI
•10060 PRINT"EXPECTED CHARACTER[3" "]";CH
R$(EC);"[3" "]WITH ASCII VALUE"EC :RETUR
N                                            MA
•11000 REM -- PUT M.L. PRINT ROUTINE INTO
 MEMORY --                                   CB
•11010 M=49376  :REM $C0E0                   OM
•11020 CS=3319 :REM CHECKSUM                 FB
•11030 READ B: IF B<0 THEN 11060             HD
•11040 POKE M,B: M=M+1: CK=CK+B              OM
•11050 GOTO 11030                            NB
•11060 IF CK<>CS THEN PRINT"ERROR IN DATA
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 59 and 60 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
 STATEMENTS STARTING AT [3"1"]20":STOP   JK
•11070 RETURN                            PH
•11080 DATA 169, 32, 200, 136, 16, 2     PD
•11090 DATA 169, 45, 32, 210, 255, 152, 1
6, 12                                    LC
•11100 DATA 138, 73, 255, 24, 105, 1, 170
, 152                                    LM
•11110 DATA 73, 255, 105, 0, 32, 205, 189
, 96                                     MI
•11120 DATA -1                           IB
```

SPRITE ROTATOR
FROM PAGE 15

BASIC PORTION

```
•0 POKE52,45:POKE56,45:CLR               PB
•1 IFPEEK(2040)<180THENPOKE2040,255      MH
•2 IFPEEK(2041)<180THENPOKE2041,254      PB
•3 IFPEEK(49152)<>76THENLOAD"ROT.ML",8,1 ND
•10 POKE53280,0:POKE53281,0:POKE53269,0:P
OKE53248,220:POKE53249,100               AO
•15 POKE53271,3:POKE53277,3              EJ
•20 POKE53251,100:POKE53250,30:PRINT"[CLE
AR][WHITE][DOWN][DOWN]SPRITE ROTATOR":PO
KE53264,2                                CK
•25 PRINT"[DOWN]DESIGNED BY JOHN FEDOR":P
OKE53287,1:POKE53288,1:GOTO120           IH
•30 A=SQR(X[UPARROW]2+Y[UPARROW]2):IFX=0T
HENB=0:GOTO55                            IA
•50 B=ATN(Y/X):IFSGN(Y)=1ANDSGN(X)=-1THEN
B=B+[PI]                                 JG
•51 IFSGN(Y)=-1ANDSGN(X)=-1THENB=B+[PI]  PO
•52 IFSGN(X)=-1ANDY=0THENB=[PI]          CL
•53 IFSGN(X)=1ANDY=0THENB=0              NO
•54 GOTO60                               CK
•55 IFSGN(Y)=1THENB=[PI]/2               IA
•56 IFSGN(Y)=-1THENB=-[PI]/2             NI
•60 D1=DE*[PI]/180:Y1=INT(.5+A*SIN(D1+B))
:X1=INT(.5+A*COS(D1+B)):RETURN           FG
•90 X1=X+11:Y1=10-Y:Q=S1*64+Y1*3+INT(X1/8
):DO=PEEK(Q)AND(2[UPARROW](7-(X1AND7))):
RETURN                                   HO
•100 IFX1<-11ORY1<-10ORX1>12ORY1>10THENRE
TURN                                     GI
•105 X2=X1+11:Y2=10-Y1:Q=S2*64+Y2*3+INT(X
2/8)                                     FE
•110 POKEQ,PEEK(Q)OR(2[UPARROW](7-(X2AND7
))):RETURN                               IM
•120 PRINT"[DOWN][DOWN]1. LOAD IN SPRITES
 SHAPES                                  FL
•125 PRINT"[DOWN]2. CHANGE SOURCE SPRITE IP
•130 PRINT"[DOWN]3. CHANGE DESTINATION SP
RITE                                     GD
•135 PRINT"[DOWN]4. ROTATE SPRITE        EK
•140 PRINT"[DOWN]5. ANIMATE SPRITES      DE
•150 PRINT"[DOWN]6. SAVE SPRITES         GL
•152 PRINT"[DOWN]7. TOUCH-UP SPRITE      GL
•153 PRINT"[DOWN]8. COPY SOURCE TO DESTIN
ATION                                    FF
•155 S1=PEEK(2040):S2=PEEK(2041)         GJ
•160 POKE198,0                           DN
•170 GETA$:IFA$<"1"ORA$>"8"THEN170       IJ
•175 A=VAL(A$):ONAGOSUB200,250,300,400,50
0,350,600,1000                           CH
•180 RUN                                 DO
•200 PRINT"[CLEAR][DOWN][DOWN]WHAT IS THE
 FILENAME (LOAD)                         OK
•202 F$="":INPUTF$:IFF$=""THENRUN        KH
•203 OPEN15,8,15,"I0                     JC
•204 INPUT#15,A,B$,C,D:IFATHENPRINTA;B$;C
;D:CLOSE15:FORI=1TO5000:NEXT:RUN         GG
•205 OPEN5,8,5,F$+",P,R                  MB
•206 INPUT#15,A,B$,C,D:IFATHENPRINTA;B$;C
;D:CLOSE15:FORI=1TO5000:NEXT:RUN         CA
•207 GET#5,A$,B$:CLOSE5                  AP
•208 C$=CHR$(0):S1=ASC(A$+C$)+ASC(B$+C$)*
256:S1=S1/64                             FM
•209 POKE2040,S1                         OD
•210 LOADF$,8,1                          IP
•220 RUN                                 GG
•250 POKE53269,1                         PO
•260 PRINT"[CLEAR][DOWN][DOWN]CURRENT SHA
PE:"S1                                   PB
•265 PRINT"[DOWN]USE +/- TO ADJUST.  <RET
URN> TO END.":POKE650,128                MF
•270 GETA$:IFA$="+"THENS1=S1+1:IFS1>255TH
ENS1=255                                 PF
•275 IFA$="-"THENS1=S1-1:IFS1<180THENS1=1
80                                       LD
•280 POKE2040,S1:PRINT"[HOME][DOWN][DOWN]
"TAB(14)S1                               NP
•285 IFA$<>CHR$(13)THEN270               AE
•290 POKE650,0:RUN                       GM
•300 POKE53269,2                         DO
•310 PRINT"[CLEAR][DOWN][DOWN]CURRENT SHA
PE:"S2                                   EC
•315 PRINT"[DOWN]USE +/- TO ADJUST.  <RET
URN> TO END.":POKE650,128                JH
•320 GETA$:IFA$="+"THENS2=S2+1:IFS2>255TH
ENS2=255                                 KB
•325 IFA$="-"THENS2=S2-1:IFS2<180THENS2=1
80                                       PP
•330 POKE2041,S2:PRINT"[HOME][DOWN][DOWN]
"TAB(14)S2                               PE
•335 IFA$<>CHR$(13)THEN320               FE
•340 POKE650,0:RUN                       KK
•350 IFS1>S2THENPRINT"NOT POSSIBLE":FORI=
1TO5000:NEXT:RUN                         PF
•353 PRINT"[CLEAR][DOWN][DOWN]WHAT IS THE
```

```
 FILENAME (SAVE)"                            BG
•354 F$="":INPUTF$:IFF$=""THENRUN            FM
•355 OPEN15,8,15,"I0":PRINT#15,"S0:"+F$      MP
•356 OPEN5,8,5,F$+",P,W                      DM
•360 A%=(S1*64)/256:B%=(S1*64)-A%*256        HB
•363 PRINT"[DOWN][DOWN]SAVING[4"."]"         PK
•365 PRINT#5,CHR$(B%)CHR$(A%);               LL
•370 FORI=S1*64TOS2*64+63:PRINT#5,CHR$(PE
 EK(I));:NEXT                                EM
•375 CLOSE5                                  FH
•377 INPUT#15,A,B$,C,D:IFATHENPRINTA;B$;C
 ;D:FORI=1TO5000:NEXT                        AP
•378 CLOSE15                                 HO
•380 RUN                                     AH
•400 FORI=S2*64TOS2*64+63:POKEI,0:NEXT       GP
•410 POKE53269,3                             JP
•415 PRINT"[CLEAR][DOWN][DOWN]WHAT IS THE
 ROTATION (IN DEGREES)";                     KI
•420 INPUTDE                                 LA
•430 PRINT"[CLEAR][DOWN][DOWN]CALCULATING
 [3"."]"DE"DEGREES[DOWN]":PRINTTAB(25)PEE
 K(2040)TAB(33)PEEK(2041)                    KP
•435 FORY=-10TO10:FORX=-11TO12:GOSUB90:IF
 DOTHENGOSUB30:GOSUB100                      PO
•445 NEXT:NEXT:PRINT"[HOME][DOWN][DOWN]"T
 AB(14)"DONE"                                MI
•450 FORX=1TO2000:NEXT:RUN                   GD
•500 PRINT"[CLEAR][DOWN][DOWN]PRESS +/- T
 O ADJUST HIGHLIGHTED SPRITE                 NO
•510 PRINT"[DOWN]PRESS <RETURN> TO SWITCH
 START/END                                   PF
•515 PRINT"[DOWN]PRESS F1 TO BEGIN.          DF
•520 POKE650,128:POKE53269,3:X=1:A1$="[RV
 SON]":A2$="[RVSOFF]                         NB
•522 PRINTA1$"[HOME][8"[DOWN]"]START:[RVS
 OFF]"S1:PRINTA2$"END  :[RVSOFF]"S2          BH
•523 POKE2040,S1:POKE2041,S2                 GN
•525 GETA$:IFA$="+"ANDX=1THENS1=S1+1:IFS1
 >255THENS1=255                              MP
•526 IFA$="+"ANDX=2THENS2=S2+1:IFS2>255TH
 ENS2=255                                    OP
•527 IFA$="-"ANDX=1THENS1=S1-1:IFS1<180TH
 ENS1=180                                    JC
•528 IFA$="-"ANDX=2THENS2=S2-1:IFS2<180TH
 ENS2=180                                    NG
•530 IFA$=CHR$(13)THENX=3-X:B$=A1$:A1$=A2
 $:A2$=B$                                    FB
•535 IFA$<>"[F1]"THEN522                     HP
•540 PRINT"[CLEAR][DOWN][DOWN]PRESS +/- T
 O ADJUST SPEED                              AJ
•545 PRINT"[DOWN]PRESS <RETURN> TO EXIT      OP
•550 POKE53269,1:X=20                        MJ
•555 FORI=S1TOS2STEPSGN(S2-S1):POKE2040,I    CG
•560 FORQ=1TOX:NEXT                          JE
•565 GETA$:IFA$="+"THENX=X+1                 GI
•570 IFA$="-"THENX=X-1:IFX<1THENX=1          NF
•575 IFA$<>CHR$(13)THENNEXT:GOTO555          KM
•580 POKE2040,S1:POKE650,0:RUN               LH
•600 PRINT"[CLEAR][DOWN][DOWN]":POKE53269
 ,5                                          EM
•606 PRINT"[HOME][WHITE]F1 - EXITS  M - M
 IRROR"                                      PG
•607 PRINT"F - FLIP  I - INVERT  CLR/HOME
 - CLEAR                                     KE
•610 SYS49158:GOSUB900                       HI
•630 LY=0:LX=0                               LN
•635 POKE2042,14                             GM
•640 FORI=896TO959:POKEI,0:NEXT              DG
•645 FORI=896TO913STEP3:POKEI,252:NEXT       BI
•650 POKE53289,7:POKE53252,LX*8+24:POKE53
 253,LY*8+82                                 IL
•655 GETA$:IFA$="[CLEAR]"THENFORI=S1*64TO
 S1*64+63:POKEI,0:NEXT:GOSUB900:GOTO655      FP
•660 IFA$="[F1]"THENRUN                      CA
•665 IFA$="F"THEN800                         BH
•670 IFA$="M"THEN850                         AC
•675 IFA$="I"THENFORI=S1*64TOS1*64+63:POK
 EI,255-PEEK(I):NEXT:GOSUB900:GOTO655        IG
•680 A=15-PEEK(56320)AND15                   LP
•685 IF(AAND1)THENLY=LY-1:IFLY<0THENLY=20    AI
•690 IF(AAND2)THENLY=LY+1:IFLY>20THENLY=0    EH
•695 IF(AAND4)THENLX=LX-1:IFLX<0THENLX=23    NE
•700 IF(AAND8)THENLX=LX+1:IFLX>23THENLX=0    GM
•705 IF(PEEK(56320)AND16)THEN650             BE
•710 A=S1*64+LY*3+INT(LX/8)                  HH
•715 B=7-(LXAND7):IF(PEEK(A)AND(2[UPARROW
 ]B))THENPOKEA,PEEK(A)AND(255-(2[UPARROW]
 B)):GOTO722                                 KN
•720 POKEA,PEEK(A)OR(2[UPARROW]B)            FJ
•722 Z=LX+LY*40+55296+160                    AO
•723 A=PEEK(Z)AND15:A1=1:IFA1=ATHENA1=12     AH
•724 POKEZ,A1                                GF
•725 GOTO650                                 PC
•800 A=S1*64:POKE252,A/256:POKE251,A-PEEK
 (252)*256:SYS49152:GOSUB900                 LP
•820 GOTO650                                 FC
•850 A=S1*64:POKE252,A/256:POKE251,A-PEEK
 (252)*256:SYS49155:GOSUB900                 BO
•870 GOTO650                                 II
•900 A=S1*64:POKE252,A/256:POKE251,A-PEEK
 (252)*256:SYS49377:RETURN                   LF
•1000 FORI=0TO63:POKES2*64+I,PEEK(S1*64+I
 ):NEXT:RUN                                  MK
```

ROT.MI

Starting address in hex: C000
Ending address in hex: C177

***Flankspeed* required for entry! See page 61.**

```
C000: 4C 09 C0 4C 48 C0 4C B7 6F
C008: C0 A0 00 B1 FB 99 40 03 F3
C010: C8 C0 40 D0 F6 A0 3C A2 21
C018: 00 84 FD 86 FE 38 A9 3C 3E
C020: E5 FD 18 65 FE A8 B9 40 23
C028: 03 85 02 18 A5 FD 65 FE D2
C030: A8 A5 02 91 FB A4 FD A6 57
```

```
C038: FE E8 E0 03 D0 05 A2 00 7C
C040: 88 88 88 C0 F0 90 D2 60 4F
C048: A0 00 B1 FB 99 40 03 C8 3C
C050: C0 40 D0 F6 A0 00 A2 00 5C
C058: A9 00 85 02 84 FD 86 FE 91
C060: 18 A5 FD 65 FE A8 B9 40 23
C068: 03 8D FF CF A2 00 8E 00 F9
C070: CF 38 A9 07 ED 00 CF A8 8F
C078: B9 AF C0 2D FF CF F0 0A 9A
C080: A5 02 AC 00 CF 19 AF C0 2E
C088: 85 02 E8 E0 08 D0 DF 38 CA
C090: A9 02 E5 FE 18 65 FD A8 45
C098: A5 02 91 FB A6 FE A4 FD 16
C0A0: E8 E0 03 D0 05 A2 00 C8 AE
C0A8: C8 C8 C0 3F D0 AA 60 01 17
C0B0: 02 04 08 10 20 40 80 A9 59
C0B8: 04 85 FC A9 A0 85 FB A2 AD
C0C0: 00 A0 00 A9 FA 91 FB C8 5C
C0C8: C0 18 D0 F9 A0 00 E8 18 0E
C0D0: A5 FB 69 28 85 FB A5 FC 28
C0D8: 69 00 85 FC E0 15 D0 E1 6D
C0E0: 60 A9 D8 85 FE A9 A0 85 18
C0E8: FD A0 00 A2 00 8C 00 CF 86
C0F0: 8E 01 CF AD 01 CF 4A 4A 63
C0F8: 4A 18 20 64 C1 A8 B1 FB F7
C100: 8D 02 CF AD 01 CF 29 07 0E
C108: AA BD 5B C1 2D 02 CF D0 5D
C110: 05 A9 0C 4C 18 C1 A9 01 9B
C118: 85 02 A9 D8 85 FE A9 A0 F0
C120: 85 FD AC 00 CF F0 10 18 39
C128: A5 FD 69 28 85 FD A5 FE 85
C130: 69 00 85 FE 88 D0 F0 AC 15
C138: 01 CF A5 02 91 FD EE 01 30
C140: CF AE 01 CF E0 18 D0 AB 05
C148: AE 00 CF E8 EA EA 8E 00 14
C150: CF A9 00 8D 01 CF E0 15 1E
C158: D0 99 60 80 40 20 10 08 1C
C160: 04 02 01 00 18 6D 00 CF BC
C168: 6D 00 CF 6D 00 CF 18 60 5B
C170: FF 00 FF 00 FF 00 FF 00 70
```

CAPTURE

FROM PAGE 58

```
•10 POKE52,56:POKE56,56:CLR:GOTO590                              OO
•20 GOSUB370:GOSUB440:POKES+5,4:POKES+1,3                        MG
•30 FORP=.TONP:SYSJY,P:IFPEEK(JS)>.THEND(
 P)=PEEK(JS)                                                     IE
•40 POKEP(P),TS+P:X(P)=X(P)+VX(D(P)):Y(P)
 =Y(P)+VY(D(P))                                                  MF
•50 P(P)=AD+X(P)+Y(P)*C:IFPEEK(P(P))<>TTT
 HEN90                                                           PA
•60 POKEP(P),TF+D(P):NEXT                                        LK
•70 SYSBM,X(.)-N9,Y(.)-N9,.:IFNPTHENSYSBM
 ,X(N1)-N9,Y(N1)-N9,N1                                           BB
•80 GOTO30                                                       DJ
•90 K=PEEK(P(P)):POKEP(P),TF+D(P):SYSBM,X
 (.)-N9,Y(.)-N9,.                                                EA
•100 IFNPTHENSYSBM,X(N1)-N9,Y(N1)-N9,1                           LM
•110 IFK=29THEN190                                               KK
•120 POKES+5,11:POKES+1,6:POKES+4,128:POK
 ES+4,129                                                        KE
•130 FORI=1TO10:POKE55665+P*21,0:FORJ=1TO
 100:NEXT                                                        CL
•140 POKE55665+P*21,1:FORJ=1TO100:NEXT:NE
 XT:POKEP(P),K:SYSCB                                             ON
•150 IFPTHENS(0)=S(0)+100*L:GOTO170                              LP
•160 IFNPTHENS(1)=S(1)+100*L                                     BK
•170 L(P)=L(P)-1:IFL(P)<1THEN280                                 CC
•180 GOSUB510:FORJ=1TO1000:NEXT:GOTO20                           EE
•190 POKES+5,4:POKES+1,3:POKES+4,128:POKE
 S+4,129                                                         KB
•200 S(P)=S(P)+5*L:OB=OB-N1:GOSUB510:IFOB
 =.THEN220                                                       LP
•210 GOTO70                                                      MH
•220 J=C:X=8:Y=4:W=24:H=5:C=2:GOSUB570                           KO
•230 SYSPL,11,5,2:PRINT"[RVSON]BONUS FOR
 PLAYER"P+1;                                                     IC
•240 B=FNR(25*L)+25:S(P)=S(P)+B                                  BK
•250 SYSPL,11,7,10:PRINT"[3">"] "MID$(STR
 $(B),2,3)" POINTS [3">"]":GOSUB510                              AD
•260 X=10:Y=11:W=21:H=3:C=4:GOSUB570:L=L+
 1:IFL>8THENL=1                                                  GL
•270 SYSPL,11,12,4:PRINT"[RVSON]PREPARE F
 OR LEVEL"L:GOTO960                                              HO
•280 IFS(0)>HSTHENHS=S(0)                                        EP
•290 IFS(1)>HSTHENHS=S(1)                                        JP
•300 X=12:Y=7:W=17:H=7:C=11:GOSUB570                             HN
•310 FORI=8TO10:SYSPL,15,I,12:PRINT"[RVSO
 N][11" "]":NEXT:GOSUB440                                        AE
•320 SYSPL,15,9,12:PRINT"[RVSON] GAME OVE
 R ":A$=STR$(HS)                                                 AO
•330 SYSPL,14,12,12:PRINT"[RVSON] HIGH: "
 RIGHT$("[4"0"]"+RIGHT$(A$,LEN(A$)-1),5)"
  "                                                              AP
•340 POKES+6,250:POKES+4,17                                      JF
•350 FORI=5TO1STEP-1:FORJ=I*5TOI*2STEP-1:
 POKES+1,J:FORK=1TO20:NEXTK,J,I                                  KL
•360 POKES+4,16:FORI=1TO4000:NEXT:GOTO700                        KA
•370 P(0)=AD+15+15*C:X(0)=15:Y(0)=15:P(1)
 =AD+C-15+(R-5)*C:X(1)=C-25:Y(1)=R-5                             AP
•380 D(0)=4:D(1)=3                                               PN
•390 IFPEEK(P(0))=29THENP(0)=P(0)+2:X(1)=
 X(1)+2:GOTO390                                                  PA
•400 IFPEEK(P(1))=29THENP(1)=P(1)-2:X(1)=
 X(1)-2:GOTO390                                                  KP
•410 RETURN                                                      CJ
•420 FORP=.TONP                                                  JL
•430 FORI=Y(P)-5TOY(P)+5:FORJ=X(P)-5TOX(P
 )+5:POKEAD+J+I*C,32:NEXTJ,I,P:RETURN                            KP
```

```
•440 FORI=0TO23:POKES+I,0:NEXT:POKES+24,1
5:RETURN                                  OG
•450 GOSUB440:POKES+1,FNR(40)+100:POKES+1
5,30:POKES+5,9                            AG
•460 POKES+4,20:POKES+4,21:RETURN         JE
•470 GOSUB440:POKES+12,13:POKES+13,0:POKE
S+5,9:POKES+6,220                         NN
•480 POKES+11,21:FORI=1TO255STEP2:POKES+1
,I:POKES+8,FNR(256):NEXT                  OF
•490 POKES+11,0:GOSUB440:RETURN           GI
•500 POKES+5,4                            CG
•510 A$=STR$(S(0)):SYSPL,1,22,11:PRINTRIG
HT$("[4"0"]"+RIGHT$(A$,LEN(A$)-1),5);     HL
•520 A$=STR$(S(1)):PRINTSPC(16)RIGHT$("[4
"0"]"+RIGHT$(A$,LEN(A$)-1),5);            LE
•530 SYSPL,13,22,1:PRINTLEFT$("[3"#"]",L(
0))" "                                    EL
•540 SYSPL,34,22,1:PRINTLEFT$("[3"#"]",L(
1))" "                                    CP
•550 SYSPL,0,24,7:PRINTLEFT$(SP$,OB+1)" "
;                                         HA
•560 RETURN                               MA
•570 FORI=0TOH-1:SYSPL,X,Y+I,C:PRINT"[RVS
ON]"LEFT$(S$,W);:IFI>0THENPRINT"[RVSOFF]
"                                         FE
•580 NEXT:SYSPL,X+1,Y+I,C:PRINT"[RVSOFF]"
LEFT$(S$,W):RETURN                        GG
•590 POKE53280,0:POKE53281,0:PRINT"[CLEAR
]"CHR$(142)CHR$(8)"[12"[DOWN]"]";         DA
•600 PRINTTAB(15)"[RED]WORKING[3"."]":DEF
FNR(X)=INT(RND(1)*X)                      HE
•610 CB=49152:CM=49205:SI=49535:BM=49333:
PL=49591:WD=49476:JY=49494:JS=49617       GM
•620 AD=16384:N9=9:TF=34:TT=32:TS=27:S=54
272:N1=1                                  AF
•630 POKES+1,130:POKES+5,9:POKES+15,30    PH
•640 POKE56334,0:POKE1,51:POKE781,9:POKE7
82,0:POKE90,0:POKE91,216:POKE88,0         GK
•650 POKE89,64:SYS41964:POKE1,55:POKE5633
4,1:POKE54272,216:POKE53272,30            IG
•660 READI:IFI>0THENI=14336+I*8:FORJ=ITOI
+7:READK:POKEJ,K:NEXT:GOTO660             CD
•670 FORI=1TO4:READVX(I),VY(I):NEXT       LD
•680 FORI=49152TO49617:READJ:POKEI,J:NEXT HO
•690 SP$=" ":FORI=1TO40:P$=P$+"[BACKARROW
]":S$=S$+" ":SP$=SP$+"]":NEXT:SYSSI       HG
•700 PRINT"[CLEAR]";:FORI=0TO23:SYSPL,0,I
,9:PRINTP$;:NEXT:PRINTLEFT$(P$,39)"[LEFT
][INSERT][BACKARROW]";                    NH
•710 X=12:Y=1:W=15:H=2:C=2:GOSUB570       HG
•720 SYSPL,17,1,2:PRINT"[RVSON]AHOY!";:SY
SPL,15,2,2:PRINT"PRESENTS:"               GG
•730 X=5:Y=6:W=30:H=5:C=5:GOSUB570        EI
•740 SYSPL,5,6,7:PRINT"[RVSON][30"[c P]"]
"                                         EE
•750 SYSPL,5,7,9:PRINT"[BACKARROW][GREEN]
[RVSON] [s U][s *][s I] [s U][s *][s I]
[c R][s *][s I][c A][c R][c S][c S] [c A
] [c R][s *][s I] [c R][s *][c S]  [RVSO
FF] "                                     GP
•760 SYSPL,5,8,9:PRINT"[BACKARROW][GREEN]
[RVSON] [s B][3" "][c Q][s *][c W] [s B]
[s *][s K] [s B] [s B] [s B] [s B][c R][
s K] [c Q][s *][3" "][RVSOFF] "           AF
•770 SYSPL,5,9,9:PRINT"[BACKARROW][GREEN]
[RVSON] [s J][s *][s K] [c X] [c Z] [c E
][3" "][c E] [s J][s *][c X] [c X][s J][
s *] [c E][s *][c X]  [RVSOFF] "          AI
•780 SYSPL,5,10,7:PRINT"[RVSON][30"[c Y]"
]"                                        PB
•790 X=7:Y=14:W=26:H=2:C=8:GOSUB570       PF
•800 SYSPL,15,14,8:PRINT"[RVSON]A GAME OF
";                                        GI
•810 SYSPL,8,15,8:PRINT"SURVIVAL OF THE Q
UICKEST"                                  EN
•820 X=12:Y=18:W=16:H=2:C=6:GOSUB570      NP
•830 SYSPL,15,18,6:PRINT"[RVSON]WRITTEN B
Y";:SYSPL,14,19,6:PRINT"MICHAEL HOYT"     ME
•840 X=4:Y=22:W=32:H=1:C=4:GOSUB570       HB
•850 POKE198,0:SYSPL,6,22,4:PRINT"[RVSON]
NUMBER OF PLAYERS  1 OR 2  ?"             AO
•860 GETA$:NP=VAL(A$)-1:IFNP<0ORNP>1THEN8
60                                        HH
•870 GOSUB450:SYSPL,4,22,12:PRINT"[RVSON]
  ENTER 1 TO 3 FOR MAZE WIDTH[3" "]"      PO
•880 GETA$:W=VAL(A$):IFW<1ORW>3THEN880    PC
•890 GOSUB450:SYSPL,4,22,14:PRINT"[RVSON]
  ENTER 1 TO 4 FOR MAZE HEIGHT  "         DC
•900 GETA$:H=VAL(A$):IFH<1ORH>4THEN900    IN
•910 GOSUB450:SYSPL,4,22,2:PRINT"[RVSON]
 ENTER STARTING LEVEL  1 TO 8  "          AJ
•920 GETA$:L=VAL(A$):IFL<1ORL>8THEN920    LH
•930 GOSUB450:FORI=0TO1:S(I)=0:L(I)=3:NEX
T                                         JH
•940 SZ=W*H:C=W*40+20:R=H*25+10:J=C       IM
•950 X=4:Y=13:W=32:H=10:C=6:GOSUB570:SYSP
L,13,17,6:PRINT"[RVSON]CREATING MAZE[3".
"]"                                       JF
•960 GOSUB470:C=J:SYSCM,C-20,R:FORI=1TO12
+2*SZ+L                                   DH
•970 A=AD+FNR(C-20)+10+(FNR(R-10)+10)*C:K
=FNR(2)                                   PJ
•980 FORJ=0TOFNR(10)+SZ/2:IFKTHEN1010     PP
•990 IFPEEK(A+J*C)<>31THENPOKEA+J*C,30:GO
TO1030                                    JH
•1000 GOTO1020                            DD
•1010 IFPEEK(A+J)<>31THENPOKEA+J,30:GOTO1
030                                       KF
•1020 J=99                                HG
•1030 NEXTJ,I:GOSUB370:GOSUB420:PRINT"[CL
EAR]":OB=3+L*3:FORI=1TOOB                 OJ
•1040 XY=16384+FNR(C-30)+10+(FNR(R-10)+10
)*C:IFPEEK(XY)=29ORPEEK(XY)=30THEN1040    AI
•1050 POKEXY,29:NEXT:FORI=0TO1:SYSPL,I*21
,19,14:PRINT"[c A][BLUE][17"[s *]"][c 7]
[c S]"                                    AE
```

```
1060 FORJ=20TO22:SYSPL,I*21,J,6:PRINT"[s
 B]"SPC(17)"[s B]";:NEXT                       AD
1070 SYSPL,I*21,23,14:PRINT"[c Z][BLUE][
16"[s *]"][c 7][c X][BLUE][LEFT][INSERT]
[s *]";:NEXT                                   HB
1080 FORI=0TO23:SYSPL,19,I,6:PRINT"[RVSO
N][c J][c L][RVSOFF]";:NEXT                    OD
1090 SYSPL,5,20,12:PRINT"PLAYER 1"SPC(13
)"PLAYER 2"                                    JP
1100 SYSPL,1,21,11:PRINT"SCORE:"SPC(15)"
SCORE:"                                        CM
1110 SYSPL,12,21,11:PRINT"LIVES:"SPC(15)
"LIVES:"                                       EA
1120 SYSPL,33,24,14:PRINT"LVL: [YELLOW]"
MID$(STR$(L),2,1);                             FE
1130 GOSUB510:GOTO20                           LP
1140 DATA27,255,255,255,255,255,255,255,
255                                            NL
1150 DATA28,255,255,255,255,255,255,255,
255                                            JM
1160 DATA29,0,60,78,223,255,255,126,60         EL
1170 DATA30,255,129,129,129,129,129,129,
255                                            GB
1180 DATA31,196,200,49,50,76,140,19,35         FN
1190 DATA35,0,0,0,24,60,126,255,0              BO
1200 DATA36,0,255,126,60,24,0,0,0              KH
1210 DATA37,0,12,60,252,252,60,12,0            AK
1220 DATA38,0,48,60,63,63,60,48,0,-1           PB
1230 DATA0,-1,0,1,-1,0,1,0                     KN
1240 DATA169,,141,21,192,141,36,192,169,
64,141,22,192,141,37,192,160                   JL
1250 DATA,162,64,185,255,255,201,29,144,
6,201,33,176,2,208,5,169                       NE
1260 DATA32,153,255,255,200,208,235,238,
22,192,238,37,192,202,224,255,208              AB
1270 DATA224,96,169,,141,70,192,169,64,1
41,71,192,169,31,160,,162                      BN
1280 DATA64,153,255,255,200,208,250,238,
71,192,202,224,255,208,242,32,170              OC
1290 DATA192,56,233,10,141,207,193,24,10
5,30,141,68,193,32,170,192,56                  HC
1300 DATA233,10,141,208,193,169,10,133,2
49,169,64,133,250,160,10,24,165                II
1310 DATA249,109,68,193,133,249,144,2,23
0,250,136,208,241,173,208,193,141              HE
1320 DATA71,193,173,207,193,168,169,32,1
45,249,136,16,251,24,165,249,109               MG
1330 DATA68,193,133,249,144,2,230,250,20
6,71,193,173,71,193,16,225,96                  ME
1340 DATA32,253,174,32,138,173,32,247,18
3,152,96,32,170,192,141,69,193                 FH
1350 DATA32,170,192,141,70,193,32,170,19
2,72,169,,133,249,169,64,133                   LP
1360 DATA250,172,70,193,240,15,24,165,24
9,109,68,193,133,249,144,2,230                 PN
1370 DATA250,136,208,241,173,69,193,24,1
01,249,133,249,169,,101,250,133                FD
1380 DATA250,104,170,189,72,193,133,251,
169,4,133,252,189,72,193,133,253               KJ
1390 DATA169,216,133,254,169,18,141,71,1
93,160,18,177,249,145,251,56,233               GL
1400 DATA27,170,189,74,193,145,253,136,1
6,240,24,165,249,109,68,193,133                II
1410 DATA249,169,,101,250,133,250,24,165
,251,105,40,133,251,144,2,230                  FP
1420 DATA252,24,165,253,105,40,133,253,1
44,2,230,254,206,71,193,16,197                 BM
1430 DATA96,100,,,,,21,5,6,7,8,2,,,,1,1        HA
1440 DATA1,1,32,170,192,208,5,173,1,220,
208,3,173,,220,41,15                           PG
1450 DATA73,15,170,189,111,193,141,209,1
93,96,,1,2,,3,,                                LI
1460 DATA,4,,,,,,,,120,169,145,141,20,3,
169,193                                        DC
1470 DATA141,21,3,169,6,141,206,193,88,9
6,206,206,193,240,3,76,49                      NG
1480 DATA234,169,6,141,206,193,173,248,5
6,72,162,1,189,248,56,202,157                  ID
1490 DATA248,56,232,232,224,8,208,243,10
4,141,255,56,208,223,32,170,192                NJ
1500 DATA72,32,170,192,72,32,170,192,141
,134,2,104,170,104,168,24,32                   OL
1510 DATA240,255,96,,,,                        EC
```

BOMBS AWAY!

FROM PAGE 39

Starting address in hex: C000
Ending address in hex: C55F
SYS to start: 49152

***Flankspeed* required for entry! See page 61.**

```
C000: 4C 6B C0 A9 01 8D 19 D0 9A
C008: A2 0B A0 00 AD 12 D0 C9 B0
C010: 5E B0 04 A6 05 A0 5E 8E 5C
C018: 21 D0 8C 12 D0 20 45 C5 A4
C020: AD 0D DC 4A 90 03 4C 31 13
C028: EA 4C BC FE 99 20 20 53 48
C030: 43 4F 52 45 20 00 20 20 BA
C038: 48 49 47 48 20 20 00 9F 39
C040: 54 4F 20 50 4C 41 59 20 5B
C048: 41 47 41 49 4E 00 50 52 4C
C050: 45 53 53 20 54 52 49 47 93
C058: 47 45 52 00 05 30 30 30 CC
C060: 30 30 30 30 30 00 4C 3D DA
C068: 30 20 00 AD 0E DC 29 FE 79
C070: 8D 0E DC A5 01 29 FB 85 3A
C078: 01 A9 38 85 FC A9 D0 85 DD
C080: FE A9 00 85 FB 85 FD A2 D0
C088: 08 B1 FD 91 FB C8 D0 F9 61
C090: E6 FE E6 FC CA D0 F2 A5 8E
C098: 01 09 04 85 01 AD 0E DC C5
C0A0: 09 01 8D 0E DC 78 A9 1B 60
C0A8: 8D 11 D0 A9 81 8D 1A D0 BB
C0B0: A9 7F 8D 0D DC A9 03 8D 8B
C0B8: 14 03 A9 C0 8D 15 03 58 38
```

```
C0C0: A0 50 A9 FF F9 E0 C4 99 94
C0C8: 80 39 88 10 F5 A9 1E 8D 66
C0D0: 18 D0 A9 00 85 06 A9 8F 28
C0D8: 8D 18 D4 A2 01 8E 2E D0 84
C0E0: A9 0F 8D 86 02 20 44 E5 F9
C0E8: A2 04 A9 C8 85 FA 86 FB 05
C0F0: A0 00 A9 EF 91 FA C8 91 12
C0F8: FA C8 A9 FA 91 FA C8 98 4F
C100: 18 65 FA 85 FA 90 02 E6 72
C108: FB A5 FB C9 08 D0 E1 A2 CC
C110: 09 A0 1E 18 20 F0 FF A9 AA
C118: 2C A0 C0 20 1E AB A2 0F 41
C120: A0 1E 18 20 F0 FF A9 36 E7
C128: A0 C0 20 1E AB A2 10 A0 C6
C130: 1E 18 20 F0 FF A9 5C A0 1E
C138: C0 20 1E AB A2 00 A0 23 49
C140: 18 20 F0 FF A9 66 A0 C0 DA
C148: 20 1E AB A9 03 85 05 A2 0C
C150: 0A A0 1E 18 20 F0 FF A9 EB
C158: 5C A0 C0 20 1E AB A9 80 2A
C160: 8D 06 D4 A9 11 8D 04 D4 E9
C168: A2 F5 86 03 A9 82 8D 1D 61
C170: D0 A9 64 85 FC A9 0A 8D 13
C178: 28 D0 A9 05 85 FB A9 00 4B
C180: 85 43 20 9D C3 A9 83 8D 85
C188: 17 D0 A9 00 A0 09 99 04 61
C190: D0 88 10 FA A0 04 A2 00 3C
C198: A9 DD 99 FA 07 8A 99 29 09
C1A0: D0 88 10 F4 A9 DE 8D FF 15
C1A8: 07 A2 00 CA 8E 15 D0 8E 20
C1B0: 1C D0 8E 0F D4 A9 80 8D C7
C1B8: 12 D4 E8 86 FE 86 02 86 1D
C1C0: 42 A0 15 B9 30 C5 99 40 42
C1C8: 03 99 55 03 99 6A 03 88 4D
C1D0: 10 F1 A2 00 8E 20 D0 A2 97
C1D8: 0D 8E F8 07 A2 DC 8E F9 7C
C1E0: 07 A9 D2 8D 01 D0 A9 20 8D
C1E8: 8D 00 D0 A9 01 8D 27 D0 77
C1F0: A9 34 8D 0F D0 A9 1C 8D 8F
C1F8: 0E D0 A2 00 BD E1 C3 9D 7B
C200: 00 37 E8 D0 F7 AC 1E D0 84
C208: A9 05 4D 5B C5 8D 5B C5 D3
C210: A9 04 85 FD A2 02 FE 03 E7
C218: D0 BD 02 D0 F0 6C A9 BC 3D
C220: 38 FD 03 D0 B0 10 98 25 A8
C228: FD F0 0B A9 00 9D 02 D0 3C
C230: 20 5C C3 4C 8A C2 BD 03 CA
C238: D0 C9 E6 D0 4D A9 D2 8D E1
C240: 03 D0 38 BD 02 D0 E9 0A D0
C248: 8D 02 D0 A9 00 9D 02 D0 C2
C250: 8D 0B D4 8D 04 D4 A9 12 DF
C258: 8D 01 D4 A9 BC 8D 05 D4 89
C260: A9 81 8D 04 D4 A9 E6 85 08
C268: A2 A5 A2 D0 FC 8D 03 D0 82
C270: A9 80 8D 04 D4 A9 11 8D 49
C278: 04 D4 AD 01 D0 18 69 0E 60
C280: 8D 01 D0 C9 FC D0 03 4C C6
C288: F3 C2 06 FD E8 E8 E0 0C 02
C290: D0 84 18 A5 FE 6D 0E D0 EE
C298: C9 1C 90 0A E9 F3 38 E9 19
C2A0: 0D 90 03 8D 0E D0 C6 02 76
C2A8: 10 30 A9 28 A0 00 85 02 E2
C2B0: AD 1B D4 2A 90 03 C8 D0 A5
C2B8: 01 88 84 FE A6 42 E8 E8 80
C2C0: E0 0A D0 02 A2 00 A9 4C 17
C2C8: 9D 05 D0 AD 0E D0 18 69 4A
C2D0: 0C 9D 04 D0 86 42 A9 92 54
C2D8: 85 96 C6 96 A5 96 8D 01 1D
C2E0: D4 E6 09 A5 09 8D 08 D4 BE
C2E8: A4 03 20 CC FF 88 D0 FA D1
C2F0: 4C 05 C2 A9 10 8D 04 D4 25
C2F8: A9 80 8D 15 D0 A9 FF 8D CD
C300: 0E D0 A9 0E 85 05 A2 03 C6
C308: A0 0C 18 20 F0 FF A9 3F C6
C310: A0 C0 20 1E AB A9 DF 85 6A
C318: A2 A5 A2 D0 FC A2 03 A0 17
C320: 0C 18 20 F0 FF A9 4E A0 ED
C328: C0 20 1E AB A9 DF 85 A2 84
C330: A5 A2 D0 FC AD 00 DC 29 F9
C338: 10 D0 B8 4C 4B C1 A9 0F E3
C340: 2D 00 DC 49 0F AE 00 D0 22
C348: C9 08 D0 03 E8 F0 0C C9 9D
C350: 04 D0 05 CA E0 1C 90 03 85
C358: 8E 00 D0 60 A4 FB 38 A2 93
C360: 06 CA 30 14 BD B0 05 69 52
C368: 00 9D B0 05 C9 3A D0 F1 82
C370: A9 30 9D B0 05 38 B0 E9 70
C378: 88 D0 E3 E8 A0 FF C8 B9 C1
C380: B0 05 D9 A0 06 F0 F7 A0 40
C388: 05 B9 B0 05 90 0B A0 05 3E
C390: B9 B0 05 99 A0 06 88 10 D8
C398: F7 C6 FC D0 43 A9 F8 8D 98
C3A0: 0B D4 A9 00 85 09 E6 43 E2
C3A8: AA A0 25 18 20 F0 FF A9 EB
C3B0: 00 A6 43 20 CD BD A5 43 2F
C3B8: 29 03 D0 12 A0 04 B9 FA 21
C3C0: 07 49 02 99 FA 07 29 02 D9
C3C8: 99 29 D0 88 10 F0 18 A5 A3
C3D0: FB 69 0A 85 FB 38 A5 03 A2
C3D8: E9 0A 85 03 A9 64 85 FC E5
C3E0: 60 00 00 00 00 15 44 14 AE
C3E8: 55 50 15 15 55 05 56 91 FA
C3F0: 15 66 54 19 6A 94 5A A9 DC
C3F8: 95 66 55 91 19 55 94 19 F7
C400: 55 A5 69 55 64 59 55 64 31
C408: 1A 55 94 16 96 90 05 96 E4
C410: 50 16 A6 94 19 AA 94 55 5F
C418: 96 54 11 45 14 00 00 05 72
C420: 00 00 00 00 00 00 00 00 20
C428: 00 00 00 00 00 00 01 10 39
C430: 00 00 40 00 01 10 00 02 83
C438: 00 00 28 00 00 28 00 02 8A
C440: AA 80 0A AA A0 2A AA A8 3E
C448: 2A AA A8 2A AA A8 2A AA 18
C450: A8 2A AA A8 0A AA A0 0A D5
C458: AA A0 00 AA 00 00 00 00 4E
```

```
C460: 00 01 55 40 15 55 54 55 0B
C468: 14 55 3F FF FC 3F 7D FC C7
C470: 0F 55 F0 03 FF C0 01 55 DF
C478: 40 05 55 50 2A AA A8 2A 0B
C480: AA A8 55 55 55 55 55 55 D3
C488: AA AA AA A6 AA 9A 55 55 1F
C490: 55 55 55 55 A6 AA 9A A6 78
C498: AA 9A 50 00 05 F0 00 0F 33
C4A0: 00 00 00 00 00 04 40 00 E4
C4A8: 01 00 00 04 40 00 02 00 EF
C4B0: 00 02 00 00 2A A0 00 2A A7
C4B8: A0 00 2A A0 00 2A A0 00 EE
C4C0: 2A A0 00 2A A0 00 2A A0 21
C4C8: 00 2A A0 00 2A A0 00 2A 88
C4D0: A0 00 2A A0 00 2A A0 00 07
C4D8: 00 00 00 00 00 00 00 00 D8
C4E0: 83 01 31 31 31 31 01 83 AE
C4E8: C7 87 E7 E7 E7 E7 81 81 DA
C4F0: 83 01 F1 E3 C7 8F 01 01 A4
C4F8: 83 01 F9 E3 E3 F9 01 83 BD
C500: C3 83 33 33 01 01 F3 F3 97
C508: 01 01 3F 03 01 F9 01 03 4B
C510: C3 83 3F 03 01 39 01 83 58
C518: 01 01 F3 E7 CF CF 87 87 A4
C520: 83 01 39 83 83 39 01 83 A2
C528: 83 01 39 01 81 F1 03 87 E4
C530: 00 00 00 AF FF FA AF FF 8A
C538: FA AF FF FA AA AA AA AA 88
C540: AA AA 00 00 00 A5 06 49 8A
C548: 01 85 06 A8 B9 5B C5 8D E5
C550: 25 D0 B9 5D C5 8D 26 D0 A7
C558: 4C 3E C3 07 00 06 0A 40 FD
```

SIDE BORDER SCROLL EDITOR

FROM PAGE 46

Starting address in hex: C000
Ending address in hex: C9E1
SYS to start: 49152

Flankspeed **required for entry! See page 61.**

```
C000: 20 1E C2 78 A9 03 85 B2 5E
C008: A9 01 85 B3 A9 00 8D FF 23
C010: 3F 85 FE 85 0A A9 7F 8D 1A
C018: 0D DC A9 1B 8D 11 D0 A9 DF
C020: 81 8D 1A D0 A9 96 8D 14 FB
C028: 03 A9 C0 8D 15 03 A9 DC C1
C030: 8D 12 D0 A9 18 85 02 A9 93
C038: FF 8D 1D D0 8D 17 D0 A9 D2
C040: 00 A8 99 00 20 99 00 21 5D
C048: 99 00 22 C8 D0 F4 A9 DC 19
C050: A0 01 99 00 D0 C8 C8 C0 AE
C058: 11 D0 F7 A9 00 A0 00 99 16
C060: 00 D0 18 69 30 C8 C8 C0 35
C068: 10 D0 F4 A9 FF 8D 15 D0 5B
C070: A9 C0 8D 10 D0 A9 01 A0 94
C078: 00 99 27 D0 C8 C0 08 D0 6C
C080: F8 A9 80 A0 00 99 F8 07 DD
C088: 18 69 01 C8 C0 08 D0 F5 63
C090: A9 37 85 01 58 60 A2 19 6C
C098: AD 12 D0 18 69 05 29 07 DF
C0A0: 09 18 EA EA EA A0 06 88 B1
C0A8: D0 FD EA EA CE 16 D0 EE F1
C0B0: 16 D0 EA EA EA EA EA EA 19
C0B8: EA EA EA EA AD 12 D0 18 0D
C0C0: 69 05 29 07 09 18 8D 11 1F
C0C8: D0 CE 16 D0 EE 16 D0 CA EF
C0D0: D0 E3 20 0F C1 20 1A C1 72
C0D8: A9 1B 8D 11 D0 A9 DC 8D 21
C0E0: 12 D0 A9 01 8D 19 D0 A6 8C
C0E8: B3 BD 29 C2 A0 27 99 00 A7
C0F0: D0 C8 C0 2F D0 F8 C6 B2 BD
C0F8: A5 B2 10 10 A9 05 85 B2 58
C100: E6 B3 A5 B3 C9 07 30 04 F8
C108: A9 00 85 B3 4C 31 EA 20 73
C110: 78 C1 C6 02 D0 03 20 BD C4
C118: C1 60 A5 C5 C9 04 D0 03 47
C120: 4C 85 C1 60 60 18 3E 02 CC
C128: 22 3E 01 22 3E 00 22 3E 4A
C130: C2 21 3E C1 21 3E C0 21 55
C138: 3E 82 21 3E 81 21 3E 80 B9
C140: 21 3E 42 21 3E 41 21 3E E1
C148: 40 21 3E 02 21 3E 01 21 6B
C150: 3E 00 21 3E C2 20 3E C1 D0
C158: 20 3E C0 20 3E 82 20 3E B6
C160: 81 20 3E 80 20 3E 42 20 81
C168: 3E 41 20 3E 40 20 3E 02 E6
C170: 20 3E 01 20 3E 00 20 60 AE
C178: A2 00 20 25 C1 E8 E8 E8 DC
C180: E0 1E D0 F6 60 78 A9 80 4A
C188: 8D 1A D0 A9 81 8D 0D DC A3
C190: A9 C8 8D 16 D0 A9 37 85 DD
C198: 01 A2 F8 9A 58 20 8A FF D2
C1A0: A9 00 8D 15 D0 8D 18 D4 38
C1A8: A9 1B 8D 11 D0 A9 06 8D 1A
C1B0: 20 D0 8D 21 D0 A9 01 8D 59
C1B8: 86 02 4C 7B E3 AD 4B C2 A8
C1C0: EE BE C1 D0 03 EE BF C1 74
C1C8: 29 3F D0 06 20 1E C2 4C 55
C1D0: BD C1 85 FB 20 04 C2 A9 62
C1D8: 31 85 01 A0 00 A2 00 B1 85
C1E0: FC 9D 06 22 E8 E8 E8 C8 27
C1E8: C0 08 D0 F3 EE E2 C1 AD B7
C1F0: E2 C1 C9 09 D0 C7 A9 37 E1
C1F8: 85 01 A9 18 85 02 A9 06 78
C200: 8D E2 C1 60 A9 00 85 FC BE
C208: A2 08 46 FB 90 03 18 69 0A
C210: 08 6A 66 FC CA D0 F3 18 8D
C218: 69 D0 85 FD 18 60 A9 30 28
C220: 8D BE C1 A9 C2 8D BF C1 A9
C228: 60 02 0A 07 0F 07 0A 12 CD
C230: 2A 20 2A 20 2A 20 13 09 2B
C238: 04 05 20 02 0F 12 04 05 8D
C240: 12 20 13 03 12 0F 0C 0C C1
```

```
C248: 20 05 04 09 14 0F 12 20 CF
C250: 2D 17 12 09 14 14 05 0E EA
C258: 20 02 19 20 0D 09 03 08 D4
C260: 01 05 0C 20 01 2E 20 07 E8
C268: 12 0F 14 14 0F 0E 20 06 F4
C270: 0F 12 20 01 08 0F 19 21 04
C278: 20 0D 01 07 01 1A 09 0E DF
C280: 05 20 0F 03 14 0F 02 05 E1
C288: 12 20 31 39 38 38 20 00 B5
C290: 00 20 20 20 20 20 20 20 71
C298: 20 20 20 20 20 20 20 20 99
C2A0: 20 20 20 20 20 20 20 20 A1
C2A8: 20 20 20 20 20 20 20 20 A9
C2B0: 20 20 20 20 20 20 20 20 B1
C2B8: 20 20 20 20 20 20 20 20 B9
C2C0: 20 20 20 20 20 20 20 20 C1
C2C8: 20 20 20 20 20 20 20 20 C9
C2D0: 00 00 00 00 00 00 00 00 D0
C2D8: 00 00 00 00 00 00 00 00 D8
C2E0: 00 00 00 00 00 00 00 00 E0
C2E8: 00 00 00 00 00 00 00 00 E8
C2F0: 00 00 00 00 00 00 00 00 F0
C2F8: 00 00 00 00 00 00 00 00 F8
C300: 00 00 00 00 00 00 00 00 00
C308: 00 00 00 00 00 00 00 00 08
C310: 00 00 00 00 00 00 00 00 10
C318: 00 00 00 00 00 00 00 00 18
C320: 00 00 00 00 00 00 00 00 20
C328: 00 00 00 00 00 00 00 00 28
C330: 00 00 00 00 00 00 00 00 30
C338: 00 00 00 00 00 00 00 00 38
C340: 00 00 00 00 00 00 00 00 40
C348: 00 00 00 00 00 00 00 00 48
C350: 00 00 00 00 00 00 00 00 50
C358: 00 00 00 00 00 00 00 00 58
C360: 00 00 00 00 00 00 00 00 60
C368: 00 00 00 00 00 00 00 00 68
C370: 00 00 00 00 00 00 00 00 70
C378: 00 00 00 00 00 00 00 00 78
C380: 00 00 00 00 00 00 00 00 80
C388: 00 00 00 00 00 00 00 00 88
C390: 00 00 00 00 00 00 00 00 90
C398: 00 00 00 00 00 00 00 00 98
C3A0: 00 00 00 00 00 00 00 00 A0
C3A8: 00 00 00 00 00 00 00 00 A8
C3B0: 00 00 00 00 00 00 00 00 B0
C3B8: 00 00 00 00 00 00 00 00 B8
C3C0: 00 00 00 00 00 00 00 00 C0
C3C8: 00 00 00 00 00 00 00 00 C8
C3D0: 00 00 00 00 00 00 00 00 D0
C3D8: 00 00 00 00 00 00 00 00 D8
C3E0: 00 00 00 00 00 00 00 00 E0
C3E8: 00 00 00 00 00 00 00 00 E8
C3F0: 00 00 00 00 00 00 00 00 F0
C3F8: 00 00 00 00 00 00 00 00 F8
C400: 00 00 00 00 00 00 00 00 00
C408: 00 00 00 00 00 00 00 00 08
C410: 00 00 00 00 00 00 00 00 10
C418: 00 00 00 00 00 00 00 00 18
C420: 00 00 00 00 00 00 00 00 20
C428: 00 00 00 00 00 00 00 00 28
C430: 00 00 00 00 00 00 00 00 30
C438: 00 00 00 00 00 00 00 00 38
C440: 00 00 00 00 00 00 00 00 40
C448: 00 00 00 00 00 00 00 00 48
C450: 00 00 00 00 00 00 00 00 50
C458: 00 00 00 00 00 00 00 00 58
C460: 00 00 00 00 00 00 00 00 60
C468: 00 00 00 00 00 00 00 00 68
C470: 00 00 00 00 00 00 00 00 70
C478: 00 00 00 00 00 00 00 00 78
C480: 00 00 00 00 00 00 00 00 80
C488: 00 00 00 00 00 00 00 00 88
C490: 00 00 00 00 00 00 00 00 90
C498: 00 00 00 00 00 00 00 00 98
C4A0: 00 00 00 00 00 00 00 00 A0
C4A8: 00 00 00 00 00 00 00 00 A8
C4B0: 00 00 00 00 00 00 00 00 B0
C4B8: 00 00 00 00 00 00 00 00 B8
C4C0: 00 00 00 00 00 00 00 00 C0
C4C8: 00 00 00 00 00 00 00 00 C8
C4D0: 00 00 00 00 00 00 00 00 D0
C4D8: 00 00 00 00 00 00 00 00 D8
C4E0: 00 00 00 00 00 00 00 00 E0
C4E8: 00 00 00 00 00 00 00 00 E8
C4F0: 00 00 00 00 00 00 00 00 F0
C4F8: 00 00 00 00 00 00 00 00 F8
C500: 00 00 00 00 00 00 00 00 00
C508: 00 00 00 00 00 00 00 00 08
C510: 00 00 00 00 00 00 00 00 10
C518: 00 00 00 00 00 00 00 00 18
C520: 00 00 00 00 00 00 00 00 20
C528: 00 00 00 00 00 00 00 00 28
C530: 00 00 00 00 00 00 00 00 30
C538: 00 00 00 00 00 00 00 00 38
C540: 00 00 00 00 00 00 00 00 40
C548: 00 00 00 00 00 00 00 00 48
C550: 00 00 00 00 00 00 00 00 50
C558: 00 00 00 00 00 00 00 00 58
C560: 00 00 00 00 00 00 00 00 60
C568: 00 00 00 00 00 00 00 00 68
C570: 00 00 00 00 00 00 00 00 70
C578: 00 00 00 00 00 00 00 00 78
C580: 00 00 00 00 00 00 00 00 80
C588: 00 00 00 00 00 00 00 00 88
C590: 00 00 00 00 00 00 00 00 90
C598: 00 00 00 00 00 00 00 00 98
C5A0: 00 00 00 00 00 00 00 00 A0
C5A8: 00 00 00 00 00 00 00 00 A8
C5B0: 00 00 00 00 00 00 00 00 B0
C5B8: 00 00 00 00 00 00 00 00 B8
C5C0: 00 00 00 00 00 00 00 00 C0
C5C8: 00 00 00 00 00 00 00 00 C8
C5D0: 00 00 00 00 00 00 00 00 D0
C5D8: 00 00 00 00 00 00 00 00 D8
C5E0: 00 00 00 00 00 00 00 00 E0
```

```
C5E8: 00 00 00 00 00 00 00 00 E8
C5F0: 00 00 00 00 00 00 00 00 F0
C5F8: 00 00 00 00 00 00 00 00 F8
C600: A9 00 8D 20 D0 8D 21 D0 A7
C608: 8D C8 C9 A9 24 A0 C8 20 7F
C610: 1E AB A9 00 85 CC 20 E4 DA
C618: FF F0 FB C9 31 F0 0F C9 C9
C620: 32 F0 12 C9 33 F0 15 C9 22
C628: 34 F0 18 4C 16 C6 A9 01 39
C630: 85 CC 4C 0D C7 A9 01 85 D3
C638: CC 4C 4B C6 A9 01 85 CC 60
C640: 4C AA C7 A9 01 85 CC 4C 48
C648: A9 C7 60 A9 5F A0 C9 20 AD
C650: 1E AB 20 C9 C9 A9 00 A2 1A
C658: 00 A9 1E 9D 28 04 8E 40 B8
C660: 03 20 E4 FF F0 F8 C9 0D 29
C668: F0 67 C9 1D F0 30 C9 2B BD
C670: F0 0E C9 2D F0 19 C9 9D D7
C678: F0 3C AE 40 03 4C 5E C6 09
C680: AE C8 C9 FE 29 C2 20 C9 96
C688: C9 AE 40 03 4C 5E C6 AE 64
C690: C8 C9 DE 29 C2 20 C9 C9 A1
C698: AE 40 03 4C 5E C6 AE 40 EA
C6A0: 03 E0 06 F0 29 A9 20 9D 0C
C6A8: 28 04 E8 EE C8 C9 A9 1E 07
C6B0: 9D 28 04 4C 5E C6 AE 40 DA
C6B8: 03 E0 00 F0 11 A9 20 9D 06
C6C0: 28 04 CA CE C8 C9 A9 1E E0
C6C8: 9D 28 04 4C 5E C6 4C 5E AE
C6D0: C6 A9 A1 A0 C9 20 1E AB 37
C6D8: 20 E4 FF F0 FB C9 31 F0 B6
C6E0: 0B C9 32 F0 12 C9 33 F0 D8
C6E8: 19 4C D8 C6 20 D2 FF A9 8A
C6F0: 05 8D FD C0 4C 00 C6 20 75
C6F8: D2 FF A9 0A 8D FD C0 4C 18
C700: 00 C6 20 D2 FF A9 0F 8D FF
C708: FD C0 4C 00 C6 A9 01 85 0A
C710: CC A9 93 20 D2 FF A9 13 C9
C718: A0 C9 20 1E AB A9 A0 8D 44
C720: E7 07 A9 00 85 FB A9 04 E7
C728: 85 FC A9 9D 8D 40 03 A9 6C
C730: 5B A0 C9 20 1E AB AD 40 CD
C738: 03 20 D2 FF A9 00 85 D4 32
C740: A9 57 A0 C9 20 1E AB 20 B5
C748: E4 FF F0 FB C9 85 F0 07 61
C750: C9 93 F0 B9 4C 2C C7 A9 42
C758: 5B A0 C9 20 1E AB A9 40 F1
C760: 20 D2 FF A9 00 85 FB A9 28
C768: 04 85 FC A9 30 85 FD A9 F5
C770: C2 85 FE A0 00 B1 FB C9 CF
C778: 00 F0 17 91 FD E6 FB 20 13
C780: 97 C7 E6 FD 20 A0 C7 4C 99
C788: 73 C7 91 FD C8 91 FD 4C F7
C790: 7D C7 91 FD 4C 00 C6 A5 1E
C798: FB C9 00 D0 02 E6 FC 60 75
C7A0: A5 FD C9 00 D0 F9 E6 FE BE
C7A8: 60 60 A9 B8 A0 C9 20 1E 74
C7B0: AB A2 00 86 CC 8E C7 C9 72
C7B8: 20 E4 FF F0 F8 AE C7 C9 E7
C7C0: C9 0D F0 27 C9 14 F0 15 93
C7C8: C9 20 30 E9 C9 5F 10 E5 EB
C7D0: AE C7 C9 9D 40 03 E8 20 FA
C7D8: D2 FF 4C B5 C7 AE C7 C9 B5
C7E0: E0 00 F0 D1 CA 20 D2 FF 42
C7E8: 4C B5 C7 AE C7 C9 E0 10 E3
C7F0: 10 B8 AE C7 C9 E0 00 F0 CB
C7F8: B1 A9 01 85 CC A9 08 A2 FB
C800: 08 A0 08 20 BA FF AD C7 01
C808: C9 A2 40 A0 03 20 BD FF 36
C810: A9 00 85 FB A9 C0 85 FC 28
C818: A2 E0 A0 C5 A9 FB 20 D8 A0
C820: FF 4C 00 C6 93 11 11 20 09
C828: 20 20 20 20 20 20 20 1F 28
C830: 12 53 43 52 4F 4C 4C 20 33
C838: 54 45 58 54 20 20 45 44 48
C840: 49 54 4F 52 20 31 2E 31 30
C848: 92 0D 20 20 20 20 20 20 A8
C850: 20 20 9A B8 B8 B8 B8 B8 C6
C858: B8 B8 B8 B8 B8 B8 B8 B8 1E
C860: B8 B8 B8 B8 B8 B8 B8 B8 26
C868: B8 B8 0D 20 20 20 20 20 87
C870: 20 20 20 1C 3C 96 31 1C 0D
C878: 3E 20 9F 2D 2D 20 1E 45 54
C880: 44 49 54 20 53 43 52 4F BA
C888: 4C 4C 20 54 45 58 54 0D 94
C890: 20 20 20 20 20 20 20 20 91
C898: 1C 3C 96 32 1C 3E 20 9F D3
C8A0: 2D 2D 20 1E 45 44 49 54 60
C8A8: 20 50 52 45 46 45 52 41 CF
C8B0: 4E 43 45 53 0D 20 20 20 48
C8B8: 20 20 20 20 20 1C 3C 96 48
C8C0: 33 1C 3E 20 9F 2D 2D 20 88
C8C8: 1E 53 41 56 45 20 53 43 CD
C8D0: 52 4F 4C 4C 0D 20 20 20 78
C8D8: 20 20 20 20 20 1C 3C 96 68
C8E0: 34 1C 3E 20 9F 2D 2D 20 A9
C8E8: 1E 51 55 49 54 20 4C 41 F8
C8F0: 52 47 45 20 53 43 52 4F 28
C8F8: 4C 4C 0D 11 11 20 20 20 21
C900: 20 20 20 20 20 1F 4F 50 5F
C908: 54 49 4F 4E 9A 20 2D 2D 58
C910: 20 9E 00 13 11 11 11 11 26
C918: 11 11 11 11 11 11 11 11 A0
C920: 11 11 11 11 11 11 11 11 A8
C928: 11 11 11 11 12 20 20 20 DE
C930: 20 20 45 4E 44 20 41 42 EB
C938: 4F 56 45 20 54 48 49 53 7C
C940: 20 4C 49 4E 45 20 57 49 4A
C948: 54 48 20 27 46 31 27 20 EA
C950: 20 20 20 20 92 13 00 12 88
C958: 20 92 00 9D 20 9D 00 93 F9
C960: 12 31 32 33 34 35 36 37 DF
C968: 0D 11 11 43 52 53 52 20 F2
C970: 4B 45 59 53 20 4D 4F 56 C0
C978: 45 20 41 52 52 4F 57 2E 98
C980: 2E 2E 2E 0D 11 27 2B 27 A2
```

```
C988: 20 41 4E 44 20 27 2D 27 18    C9B8: 9E 93 11 11 46 49 4C 45 2E
C990: 20 54 4F 20 43 48 41 4E 8F    C9C0: 20 4E 41 4D 45 3A 00 00 3D
C998: 47 45 20 43 4F 4C 4F 52 C5    C9C8: 00 A0 00 B9 29 C2 99 00 A8
C9A0: 00 0D 11 11 53 45 4C 45 F9    C9D0: D8 C8 C0 07 D0 F5 60 BD 1F
C9A8: 43 54 20 53 50 45 45 44 D2    C9D8: EA 80 7F 75 48 40 55 35 4C
C9B0: 20 28 31 2D 33 29 3A 00 ED    C9E0: A8 FF 89
```

REVIEWS

Continued from page 56

leted if you wish a naked look, but whether or not they are necessary depends on your level of knowledge.

With the cursor keys, you can "slew" the screen in increments to turn your viewpoint through a full 360 degrees, and you can also adjust elevation to any point between the zenith (straight up) or nadir (straight down). At nadir, you'll find yourself looking through the earth, with heavenly bodies in place on the other side of the world, and you'll also find yourself looking at your feet. (One shoelace is untied.)

As you perform these moves, using either the cursor keys or a joystick, the legend in the right screen border updates itself to show the new figures for elevation, azimuth, right ascension, and declination. Also displayed here are the date and time selected, the timezone from which you are viewing, the clock rate, and angle of view. Just below these items there is an onscreen reminder as to how the function keys are used.

By moving the crosshair cursor about the sky, it is possible to center on an object and then touch the Inform (F7) key. A message regarding the object will appear in the lower border and will not only identify the object, but will tell what is known regarding magnitude, distance from earth, and so on.

Should you know what you wish to see, a faster way is to use the Find (F5) key. Cycle through the moon, planets, and Halley's comet, and you'll find a list of major and minor constellations. Select the one you want, press Return, and the Sky screen will reconfigure to show that object at screen center.

If you've picked a constellation, it will add to the fun to select Deep Sky with the Options key (F3). Here, far galaxies will appear as whorls and spirals. A good example is Andromeda. Though the cursor will center on the constellation, to the left of center you'll see the Andromeda galaxy (M31), looking like a saucer turned on end. Now magnify your view by decreasing the viewing angle from a default 72 degrees to 9 degrees. The result is the same as looking through a telescope with variable magnification, and now you'll see not only M31 but the two satellite galaxies.

Screens discussed so far are Sky and Map, accessed through the Fl key. There is a third called Chart, and this displays the stars and constellations on a "celestial sphere," with demarcation lines showing declination and right ascension. In this mode the stars are black on a white background, and this makes it a mode perfect for printing. Though only a portion of the sky is shown, as in Sky mode, it should be helpful to make a printout for tonight and then take the map and a penlight outside to help you in identifying the sights you'll see.

Some you won't see are the historical events made possible by setting the calendar for past dates. Halley's comet is visible only during the period of its approach to earth a few years ago. The documentation gives guidelines for other historical stellar events such as alignments of planets, solar eclipses, and even the "star" of Bethlehem–now thought to be a planetary conjunction.

For other periodic events, such as transits of the sun by Mercury and Venus and both solar and lunar eclipses, the *Sky Travel* clock can be set to an accelerated rate so that the events happen in less time.

In the original version of *Sky Travel* were a few inside jokes, and one magazine conducted a kind of forum for several issues as new treasures were found. All were ground objects you wouldn't expect but would be pleasantly surprised to find: set the coordinates for St. Louis, Mo. and you can slew around until you spot the Gateway Arch. Select New York City and you'll see the Empire State Building. (One report from those days had it that King Kong was clinging to the side of the building. He is not there now, perhaps because lately he hangs out at the World Trade Center.)

The documentation for *Sky Travel* is excellent, giving you a good grounding in the rules and laws of astronomy, explanations of planetary ephemeris, discussions as to why the Julian calendar is used, and more. All of it is presented in very readable form.

As to the program itself, it is a marvel, and will be useful to anyone who has ever looked at the night sky in wonder. It is well worthy of reissue.

Microillusions, 17408 Chatsworth Street, Granada Hills, CA 91344 (phone: 800-522-2041 or 818-360-3715).

–Ervin Bobo

...COMING IN THE NOVEMBER ISSUE OF *AHOY!* (ON SALE OCTOBER 11)...

COMPILING STRINGS
IMPLEMENTING VARIABLES IN A BASIC COMPILER

ISLAND QUEST
A C-64 SEA HUNT

AGOG WITH GAMES!
EXPANDED ENTERTAINMENT COVERAGE


THE NAME OF THE GAME IS SURVIVAL.
PLATOON™
In Vietnam, an American soldier soon learned that there were no winners in this deadly game. To survive each mission with his morale and sanity intact, and return to base safely, was all he could hope for. This was captured in the award-winning film, Platoon.
In Data East's Platoon, you will experience the full impact of the film as you lead your platoon into the jungles, villages, bunkers, foxholes, and underground tunnels of war-torn Southeast Asia.
There you will encounter guerilla fighters, snipers, booby-trapped trip wires, armed patrols, and napalm air strikes. You must pick up food, medical supplies, and ammunition along the way.
At times, the odds may seem insurmountable. But don't think about winning – to survive is enough.
FOR COMMODORE 64/128™ AND AMIGA,™ ATARI® ST, APPLE II SERIES,® AND IBM.®
DATA EAST
DATA EAST USA, INC.
470 NEEDLES DR., SAN JOSE, CA 95112. (408) 286-7074.
LICENSED BY
ocean®
GAME DESIGN © 1987 OCEAN SOFTWARE LTD. PLATOON: TM & © 1986 HEMDALE FILM CORP. ALL RIGHTS RESERVED. COMMODORE, APPLE, ATARI, IBM, AND OCEAN ARE REGISTERED TRADEMARKS OF COMMODORE ELECTRONICS LTD., APPLE COMPUTERS, INC., ATARI CORP., IBM CORP., AND OCEAN SOFTWARE LTD., RESPECTIVELY.